Automated Problem List Generation from Electronic Medical Records in IBM Watson

Murthy Devarakonda, Ching-Huei Tsou
IBM Research and Watson Group
Yorktown Heights, NY 10598
{mdev,tsou}@us.ibm.com

Abstract

Identifying a patient's important medical problems requires broad and deep medical expertise, as well as significant time to gather all the relevant facts from the patient's medical record and assess the clinical importance of the facts in reaching the final conclusion. A patient's medical problem list is by far the most critical information that a physician uses in treatment and care of a patient. In spite of its critical role, its curation, manual or automated, has been an unmet need in clinical practice. We developed a machine learning technique in IBM Watson to automatically generate a patient's medical problem list. The machine learning model uses lexical and medical features extracted from a patient's record using NLP techniques. We show that the automated method achieves 70% recall and 67% precision based on the gold standard that medical experts created on a set of de-identified patient records from a major hospital system in the US. To the best of our knowledge this is the first successful machine learning/NLP method of extracting an open-ended patient's medical problems from an Electronic Medical Record (EMR). This paper also contributes a methodology for assessing accuracy of a medical problem list generation technique.

Introduction

In clinical care, a patient's medical problem list describes diagnosed diseases that require care and treatment as well as key medical symptoms that have not been diagnosed yet. Since the publication of Dr. Weed's seminal paper on problem-oriented medical record (POMR) in 1968 (Weed, 1968), organizing medical records around the problem list has come to be accepted as an important goal. However, creating and maintaining an accurate problem list has proved to be quite difficult. While the modern EMR systems improved patient data collection, the problem list maintenance was still left to manual efforts. Achieving the goals of the meaningful use initiative and efficient medical care requires automated methods for generating the problem list and an assessment methodology of their accuracy.

Since winning the American TV quiz game called Jeopardy! in 2011 (IBM Research, 2012), we have been adapting IBM Watson to the medical domain so that the technology can help physicians and clinicians provide improved care (Ferrucci, et al., 2013). A recent adaptation of IBM Watson is its application to Electronic Medical Records Analysis (EMRA), of which the automated problem list generation described here is a key component.

The IBM Watson problem list generation starts with identification of a large pool of medical disorders mentioned in the clinical notes of a patient's EMR and then whittles down this larger list to a smaller accurate problem list using NLP and machine learning.

To evaluate the accuracy of the Watson method, we asked medical experts to create a gold standard using 199 EMRs acquired from Cleveland Clinic under an IRB protocol for the study. We set aside a test set of 40 random EMRs from the gold standard and used them to assess the accuracy of the Watson method. The Watson method recall is 70% and the precision is 67%, with an F1 score of 0.69. The key contribution of this paper is a practical and accurate automated method of generating an open-ended problem list from a longitudinal EMR.

Background

Many hospitals and physicians are now routinely using EMRs as a part of patient care. An EMR typically contains several plain text documents known as clinical notes that are written as a result of patient contacts. An EMR also contains several sections of semi-structured data such as medications ordered, laboratory test results, and procedures conducted.

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

An EMR usually contains a section for medical problems to be entered and maintained by physicians and clinical staff. In spite of its clear value, however, the problem list section is rarely well maintained and almost always ignored by physicians (Campbell, 1998) (Meystre & Haug, 2008) (Holmes, 2011 Feb) (Holmes, 2011 Mar). Often stated reasons include: lack of proper support from EMR systems, lack of clarity of what goes on the list and when a problem (if at all) comes off of the list, multiple authors, and multiple and often contradictory uses of the list. Perhaps the fundamental reason is that it is a knowledge and time intensive task requiring significant investment of an expert's time, which is always in short supply. Therefore, it is a task that requires an automated and intelligent solution.

Related Work

There are several efforts to define better coding systems to represent medical problems (Campbell & Payne, 1994) and there is even more recent activity to define a new coding system based on a subset of SNOMED CT (US National Library of Medicine, 2014). However, the closest work, i.e. that of automation of problem list generation, is reported in a series of papers by Meystre and Haug (Meystre & Haug, 2005) (Meystre & Haug, 2006) (Meystre & Haug, 2008).

The main result from the work of Meystre and Haug is the identification of a patient's medical problems in a specific domain (e.g. cardiovascular patients) from a list of *apriori* identified problems using simple NLP techniques and an assessment methodology of its accuracy. What is common between their method and ours is that both analyze plain text clinical notes in patient medical records. However, the key difference is that our goal is an open-ended problem list generation rather than limiting it to a list of diseases specific to a domain or a patient population. This difference is critical because the problem that is most important for patient care may be outside the known domain.

Because our goal is an open-ended problem list generation, our method cannot simply search for disease terms (and their semantic equivalents) from a list as in Meystre and Haug. For example, our approach is not that of assessing if Myocardial Infarction is a problem for a patient, but that of assessing if any of the diseases, syndromes, symptoms, findings, or procedures appearing in a patient's EMR should be in the patient's medical list. As our other method is an open-ended problem list extraction and it is necessary to apply advanced AI techniques.

Watson Problem List Generation

As shown in Figure 1, Watson problem list generation begins with an automated identification of medical concepts in all parts of an EMR – both in the plain text clinical notes and in the remaining semi-structured clinical data. Terms representing medical concepts are assigned one or more Concept Unique Identifiers (CUIs) from the UMLS metathesaurus (US National Library of Medicine, 2009). CUI mapping allows reasoning about medical concepts; for example, it becomes possible to recognize that the phrases HTN, Hypertension, and High Blood Pressure all refer to the same disease. Also, they can be categorized into semantic groups, e.g. as Disorders, Chemicals & Drugs, Procedures, etc. Each of these groups is further subcategorized, for example, Disorders are sub-grouped as Diseases or Syndromes, Signs or Symptoms, Findings, and others.

Mapping terms to CUIs is, in itself an interesting research task. Both the CUI space and the term space are large and the mapping is many-to-many. Using the context around a term is often necessary to obtain a CUI that more accurately represents the concept. So, in Watson, in addition to the standard NLP and UMLS lookup, we use additional contextual and sentence structural information to obtain a better mapping. (The details are beyond the scope of this paper.) A numerical score indicates how confident we are that a CUI represents the original term, and it is used as a feature in problem list generation. Once one or more CUIs for a concept are identified, the CUIs are then mapped to a SNOMED CT CORE (US National Library of Medicine, 2014) concept. If there is no exact match, we climb the UMLS hierarchy until the closest parent that has a SNOMED CT CORE concept is reached.

For a typical EMR, usually a few hundred candidate problems are identified after the first step. When compared to the final list, the problems generated in the first step would have high recall (>90%) but poor precision (<10%). The subsequent steps attempt to improve precision of the problem list without substantial loss of recall.

In the second step, the method produces feature values for the lexical and clinical features of the machine learning model. An example lexical feature is the TF-IDF of a po-

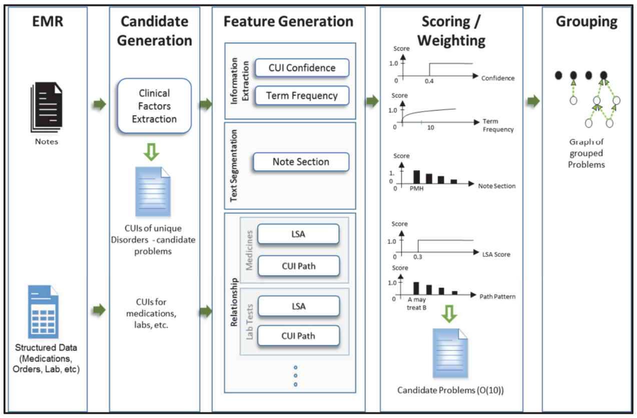


Figure 1 Watson Problem List Generation Overview

tential problem. An example clinical feature is the probability that at least one of the patient's active medications is a treatment for this potential problem. Key features are listed and briefly described in Table 1 and further discussed later in the paper. We used the Alternating Decision Tree (Freund & Mason, 1999) technique for its accuracy and clarity of the decision process.

The reason for representing the generated problems as the SNOMED-CT CORE concepts is that it is a result of the efforts to define a standard vocabulary for documentation and encoding of clinical information at a summary level, such as the problem list, discharge diagnosis, or the reason for an encounter (US National Library of Medicine, 2014).

Features

Longitudinal EMRs are a rich source of information and a lot of data can be extracted. Assembling this extracted data into features with the appropriate type and level of aggregation is a practical question and crucial to success. Key features that contributed to our final results are listed and briefly described in Table 1. Each feature category is explained in more detail in this section.

Lexical Features

Standard TF-IDF formulation is used, where TF is normalized using the maximum frequency of any term in the document. TF-IDF reflects how important a term is to a document in a corpus. In our case, a term is a candidate problem. Depending on the goal, a document can be a note or an EMR. When generating the problem list for a patient, an EMR is a document and the entire collection of EMRs is our corpus. When deciding which note is more relevant to a selected problem, the note becomes the document and an EMR becomes the corpus. For the problem list generation, IDF is calculated using the entire de-identified EMR collection that we have.

Table 1 Description of the Key Features of the Model

Feature	Description
Freq Prob	Frequent problems (top 25 of most frequently diagnosed problems).
Diagnosed	Has ever been diagnosed (structured data)
S_PMH	Problem is found in the past medical history section
Med Score	Problem is related to any active medication
TF-IDF	Term Frequency and Inverse document frequency
TF (1st section)	TF; problem appears in the 1st section of the EMR
TF (A&P)	TF; problem appears in the "assessment and plan" section of the EMR
PCA	Probability of Concept for a given term (covered text)
Freq Prob (CORE)	SNOMED CORE usage: the average usage percentage among all institutions
1st Date	Date (normalized) the problem is first occurred
TF	Term-frequency
TF (Recent)	TF; prob. appears in the recent (last 3 mo.) notes
TF (Recent, RoS)	TF; problem appears in the "Review of System" section in the recent (last 3 months) notes

Unlike a normal text document, an EMR is longitudinal record and therefore, more recent notes are likely to better represent the patient's medical problems. Also, each note in the EMR has implicit sections and so a concept (e.g. hypertension) appearing in different sections (e.g. family history vs. assessment and plan) may have significantly different meanings. Because of this, in addition to calculating TF at the EMR level, TF is also calculated for each note section and for a few different time periods.

Medical Features

Terms in the EMR semi-structured data are also mapped to UMLS CUIs so that we can use the UMLS relations. Medications turn out to be one of the most important features, whereas we saw no benefit from the lab tests and procedure orders. The first reason is that the medication names are relatively standardized, even while mixing the generic and brand names, and a UMLS CUI can be reliably found. Conversely, labs and procedures are often specified in institution specific abbreviations instead of CPT codes and LOINC codes, and are therefore harder to accurately map to UMLS concepts. Second, medications are prescribed to treat problems, while lab tests and procedures are often ordered to diagnose a problem and extensive domain knowledge is needed to interpret their results. The relation between a problem and a medication is derived from a weighted confidence score obtained from distributional semantics (Gliozzo, 2013) and UMLS relationships.

Frequency Features

Frequency of a problem can be thought of as the prior probability that the patient may have it. Two sources of frequency are used in our experiments. The first is the SNOMED CORE usage (US National Library of Medicine, 2014), which represents the frequency in a broad population. The second is calculated using all diagnosed problems (as ICD-9 codes) in our collection of EMRs, which represents the frequency in this particular institution.

Structural Features

The concept "diabetes mellitus" appearing in the assessment and plan (informal) section in a patient's progress note is a much stronger indicator that the patient has the disorder than the same concept detected in the family history section in a nursing note. Since notes are in plain text and note metadata is optional, the structures have to be learned. Informal sections of a note are detected with regular expressions and heuristic rules. Note types are learned using a Maximum Entropy classifier with the available metadata, and several medical and lexical features from the note text.

Temporal Features

The span of an EMR varies from a single day to several decades. Most temporal features in our experiments are normalized to prevent bias towards longer EMRs, but absolute value is also used to define certain features, e.g. note *recency*, where the recency is defined as the number of days from the latest patient contact.

Temporal data is used in three ways. First, it is used as features directly. Temporal features considered include the first/last mention of a problem, and the duration of a problem. Second, it is used to align semi-structured data and structure data, e.g. a medication prescribed before a problem is mentioned in a note is not considered as evidence to the problem. Third, temporal data is used to divide notes into bins on the timeline so that frequency can be counted by intervals, e.g. TF in recent notes vs. TF in earlier notes.

Model

We construct problem list generation as a binary classification problem, i.e., for each candidate problem in an EMR, the task is to classify it as a problem or a non-problem. We initially used a SVM model (Cortes & Vapnik, 1995) (Chang & Lin, 2011) with linear kernel, but soon favored more human interpretable models. As the gold-standard is expensive to get and the training data is limited, knowledge coming from domain experts and error analyses become critical to success – and both benefit from models that output human understandable decision process. Decision tree and association rules based classifiers generate models close to the way medical experts think, at the cost of usually lower accuracy. We observed performance similar to our earlier SVM model by using alternating decision tree (ADT) (Freund & Mason, 1999), which outputs an option tree but has its root in boosting. The basic implementation of ADT (Hall, et al., 2009) uses a decision stump as the base learner and adaptive boosting to grow the tree iteratively. During a boosting iteration, ADT adds a splitter node and corresponding prediction nodes to extend one of the existing paths in the tree. The scores associated with the prediction nodes are obtained from the rules.

Model parameters are selected using 10-fold cross-validation. The number of iterations of ADT is set to 30 (from the ROC and the Recall-Precision graphs), and the score threshold is set to 0.85, to maximize the training F1 score. A subset (some branches are omitted after the first two levels) of the tree generated by our model is shown in Figure 2.

Being a boosted algorithm, ADT picks the strongest weak learner first, which is, in our experiments, the problem diagnosed frequency. In each iteration, the misclassified instances are given a larger weight while correctly classified problems are given reduced weight – so the model consequently focuses on classifying the hard instances correctly. The top level features in Figure 2 are all intuitive – but it is important to understand that they are not necessarily the most important features to determine whether a candidate problem is, in fact, a patient's active problem – they simply work better for the easy instances. Some less intuitive features also shed some light on how EMRs are written. For example, it is a positive indicator, if a problem appears in the first section, regardless of what the section is. This is because many EMRs start by stating the patient's active concerns. Another example is the first mention date because a patient's past medical history are often carefully documented in his/her first visit to the hospital.

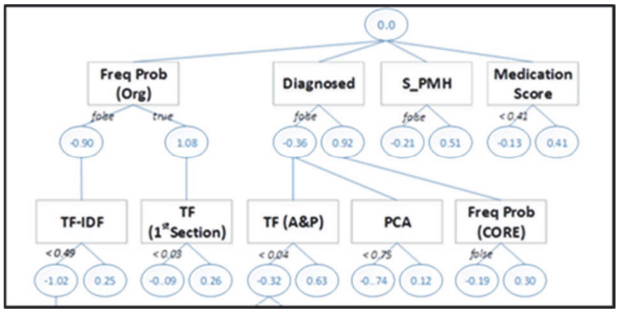


Figure 2 First two levels of our ADT model

Gold Standard and Accuracy Analysis

To the best of our knowledge, there is no publicly available gold standard for problem lists, so we developed a gold standard of our own. The process involved two fourth year medical students studying 199 EMRs and each creating a problem list for each of the EMRs. An MD/PhD then reviewed and adjudicated any differences between the students' problem lists for each EMR. The gold standard problem list is coded using the CORE subset of SNOMED CT. Often there are more than one code that is a good match to a medical problem. In these cases, all codes are considered acceptable codes for the problem.

We compared the Watson generated problem lists with the problem lists in this gold standard. If a problem appears on both problem lists of an EMR, then it is a true positive. If a problem appears in the gold standard for an EMR, but not in the Watson's generated list for the EMR, then it is a false negative. If a problem appears in the Watson generated list but not in the gold standard for the EMR then it is a false positive. However, in the case a problem has more than one acceptable code, matching any one code counts as one TP. In a stricter analysis that we report separately, if Watson reports some or all of the acceptable codes as separate problems, we consider only one as TP and the others are considered as FP.

Results

In this section we present results that provide not only the method accuracy, but also the insights that characterize the challenges of problem list generation.

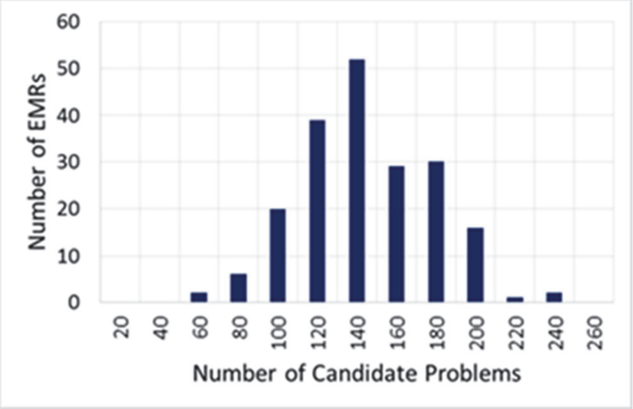


Figure 3 Distribution of the number of candidate problems

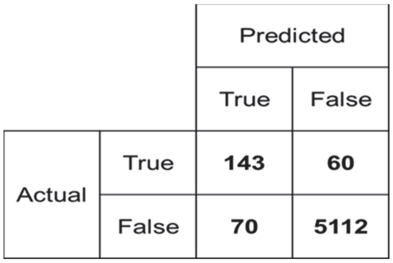
Candidate problems

Figure 3 shows a distribution of the number of candidate problems generated per EMR (across all 199 EMRs). We see a nearly normal distribution, with an average of 135 candidate problems and a standard deviation of 33. The machine learning model reduces these candidate problems to an average of 9 predicted final problems, a reduction by over 93%.

Confusion Matrix and F Scores

Table 2 shows the confusion matrix for the model on a test set of 40 EMRs from the 199 gold standard set. The remaining 159 EMRs are used to train the model. As we are analyzing the accuracy in predicting each problem here, EMRs with a larger number of problems (as per the gold standard) have a more significant influence on the results than the EMRs with fewer problems.

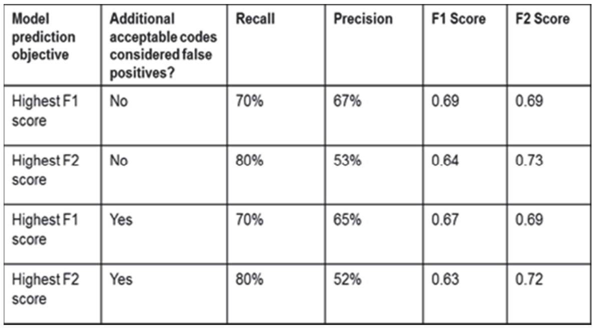
Table 2 The Confusion Matrix showing Watson accuracy based on the 40 test EMRs



		Predicted	
		True	False
Actual	True	**143**	**60**
	False	**70**	**5112**

The summary of the accuracy analysis, presented in Table 3, shows recall of 70% and precision of 67%, resulting in an F1 score 0.69 for the Watson method. This is achieved with a candidate filtering threshold of 0.85 in the machine learning model. For patient care, higher problem list recall may be more important than precision (i.e. don't miss a problem even if the list is a bit more noisy) and in that case by selecting a lower threshold (0.70) we can achieve recall of 80% and precision of 53%, resulting in an F2 score 0.73.

Table 3 Summary of the Accuracy Analysis



Model prediction objective	**Additional acceptable codes considered false positives?**	**Recall**	**Precision**	**F1 Score**	**F2 Score**
Highest F1 score	No	70%	67%	0.69	0.69
Highest F2 score	No	80%	53%	0.64	0.73
Highest F1 score	Yes	70%	65%	0.67	0.69
Highest F2 score	Yes	80%	52%	0.63	0.72

In a stricter analysis of Watson accuracy, where we considered the additional acceptable codes it generates (beyond the first one) as FP, the correspondingly highest F1 score is 0.67 and the highest F2 score is 0.72.

Most frequent problems

Figure 4 shows the 15 most frequently occurring problems and their frequency in the gold standard. Juxtaposed against them, the Figure also shows that the Watson prediction closely follows the gold standard, and so we may conclude that Watson performances well for frequently occurring problems. However, lower back pain provides an interesting contrast: It is a challenge for our model because there is usually no medication for it and medical experts used somewhat non-specific reasons, such as the severity and there not being another problem that explains the finding, for including it in the gold standard.

Features with the strongest contribution

Which features had the strongest positive contribution for correct predictions in the Watson method? Figure 2 shows the top two levels of the ADT for the model used in Watson. Problem occurrence frequency, whether it is in the diagnosis codes, S_PMH (whether the problem is in the previous medical history part of a note), and the fact that the patient is on a medication that may treat the problem have the strongest influence on correct predictions.

Discussion

Does this result generalize? The content and the format of the EMRs we used here for training and testing are neither unique nor customized for this application. The feature set, the methodology for extracting feature values and for calculating feature scores, and the machine learning tech-

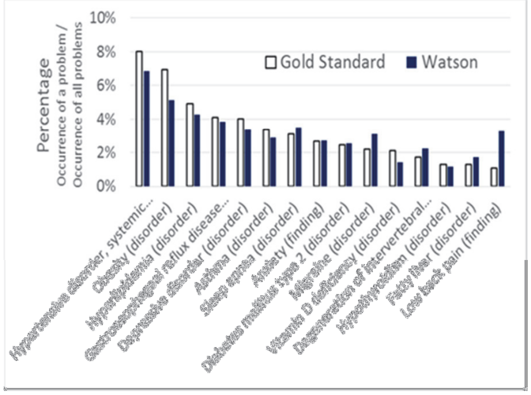


Figure 4 ***Top 15 frequent problems in the gold standard and their percentages from the Watson generated problem lists.***

niques used here are broadly applicable to EMRs from any hospital system and to data from most EMR vendor products. We believe the method and the results will therefore generalize very well.

What can be done to improve the accuracy further? This is a subject of our ongoing research, but our initial analysis of incorrect predictions points to a few areas of possible improvement. First, better tuned, broad scoped (i.e. sentence and paragraph level) negation detection would avoid several false positives. Second, while our method is suited for extracting previously diagnosed diseases, the gold standard contains undiagnosed diseases that are implied by symptoms present in EMRs. Third, mental disorders require improved support for identification and similarity reasoning. De-identification of clinical notes, abbreviations, and generally informal style of writing notes (i.e. many cut-and-paste and non-sentences) are the other causes of inaccuracy. We are continuing to improve the accuracy of our method.

Conclusion

Prior to this work, what was well established, in prototype applications, software libraries, and even in commercial products, is the ability to extract medical concepts representing diseases, syndromes, signs, symptoms, etc., from any medical text and EMR data. It was also shown, with a prototype implementation and formal assessment approaches, that a known list of problems in a certain narrow medical domain, such as the cardiovascular diseases, can be identified.

This emerging application demonstrated that an open-ended medical problem list can be generated from an EMR with high accuracy. NLP and machine learning techniques can be successfully applied to EMR contents to generate these medical problems. This application can be used to automate the management of problem lists, and as such, contributes to improved patient care.

Acknowledgements

We thank the physicians and IT staff at Cleveland Clinic who guided definition of the requirements for this application and provided de-identified EMRs under an IRB protocol for the study. We also acknowledge the groundbreaking work of our Watson team colleagues, past and present, which made this application possible.

References

Campbell, J. R., 1998. *Strategies for Problem List Implementation in a Complex Clinical Enterprise.* Lake Buena Vista, FL, American Medical Informatics Association (AMIA).

Campbell, J. R. & Payne, T. H., 1994. *A Comparison of Four Schemes for Codification of Problem Lists.* San Francisco, American Medical Informatics Association (AMIA).

Chang, C.-C. & Lin, C.-J., 2011. LIBSVM : a library for support vector machines, 2:27:1--27:27,. *ACM Transactions on Intelligent Systems and Technology,* 2(3), pp. 27:1 - 27:27.

Cortes, C & Vapnik, V.,1995. Support-vector networks. *Machine Learning* 20(3): pp. 273-297

Ferrucci, D. et al., 2013. Watson: Beyond Jeopardy!. *Artificial Intelligence,* pp. 93-105.

Freund, Y. & Mason, L., 1999. *The Alternating Decision Tree Algorithm.* San Francisco, Proc. of the 16th Int'l Conf on Machine Learning.

Gliozzo, A., 2013. *Beyond Jeopardy! Adapting Watson to New Domains Using Distributional Semantics.* [Online]: https://www.icsi.berkeley.edu/icsi/sites/default/files/events/talk_20121109_gliozzo.pdf [Accessed 18 04 2014].

Hall, M. et al., 2009. Mark Hall, Eibe Frank, GeoffreThe WEKA Data Mining Software: An Update. *SIGKDD Explorations,* 11(1).

Holmes, C., 2011 Feb. The Problem List Beyond Meaningful Use, Part I. *Journal of American Health Information Management Association,* 81(2), pp. 30-33.

Holmes, C., 2011 Mar. The Problem List beyond Meaningful Use, Part 2. *Journal of American Health Information Management Association,* 81(3), pp. 32-35.

IBM Research, 2012. This Is Watson. *IBM Journal of Research and Development,* 56(3.4), pp. 1:1 - 1:15.

Meystre, S. & Haug, P. J., 2005. Automation of a problem list using natural language processing. *BMC Medical Informatics and Decision Making* , 5(1), pp. 1-16.

Meystre, S. & Haug, P. J., 2006. Natural language processing to extract medical problems. *Journal of Biomedical Informatics,* Volume 39, pp. 589-599.

Meystre, S. M. & Haug, P. J., 2008. Randomized controlled trial of an automated problem. *International Journal of Medical Informatics,* Volume 77, pp. 602-612.

US National Library of Medicine, 2009. *UMLS Reference Manual.* [Online] http://www.ncbi.nlm.nih.gov/books/NBK9675/ [Accessed 15 04 2014].

US National Library of Medicine, 2014. *The CORE Problem List Subset of SNOMED CT.* [Online] http://www.nlm.nih.gov/research/umls/Snomed/core_subset.html [Accessed 16 Sep 2014].

Weed, L. L., 1968. Medical Records That Guide and Teach. *New England Journal of Medicine,* March.pp. 652-657.