Tracking medication information across medical records

Juan Eugenio Iglesias, Krupa Rocks, Neda Jahanshad, Enrique Frias-Martinez, Lewellyn P. Andrada, Alex A.T. Bui, PhD
University of California Los Angeles, Los Angeles, CA

Abstract

A patient's electronic medical record can consist of a large number of reports, especially for an elderly patient or for one affected by a chronic disease. It can thus be cumbersome for a physician to go through all of the reports to understand the patient's complete medical history. This paper describes work in progress towards tracking medications and their dosages through the course of a patient's medical history. 923 reports associated with 11 patients were obtained from a university hospital. Drug names were identified using a dictionary look-up approach. Dosages corresponding to these drugs were determined using regular expressions. The state of a drug (ON, OFF), which determines whether or not the drug was being taken, was identified using a support vector machine with features based on expert knowledge. Results were promising: prec. ≈ recall ≈ 87%. The output is a timeline display of the drugs which the patient has been taking.

Introduction

Patients with chronic or complicated diseases are often on multiple medications to treat the symptoms and underlying problems. These patients are typically seen by an assortment of physicians, usually a primary care physician and a few different specialists, with each visit to a healthcare provider possibly generating several medical documents [1]. The result is a set of reports from heterogeneous sources with no standards for listing medications, and typically no single source for drug information. This bevy of reports leads to the dangerous prospect of drug complications: with upwards of 100,000 fatal medical errors, including adverse drug events, occurring annually in the United States (US) [2]; methods tackling such issues are imperative.

Centralized pharmacy databases [3] are becoming common place among large healthcare centers. However, these systems are limited in the tracking of patient medications for two reasons. First, the patient must get the prescription filled at this location; otherwise, the database will never be notified that the patient is taking the drug. The second problem is that the patient filling the prescription does not necessarily indicate that the individual was compliant with the regimen [4].

While the latter problem is hard to solve for practical reasons, the former can be tackled through use of electronic medical records (EMR). The goal of the EMR is to allow healthcare providers to centrally document all information associated with a patient's health, irrespective of where this information was generated. EMRs, which are not yet a full reality in the United States but a focus of much research [5-7], would arguably help provide better [8] and more affordable [9] healthcare.

Most of the medication information in the electronic medical record is given in free-text reports, such as in admission and discharge summaries, or specialist consults. The information on medication is mostly in the form of historic references (*e.g.,* "the patient was prescribed drug *X*"), changes in dosage (*e.g.,* "drug *X* will be tapered off…") and planning (*e.g.,* "we are going to give the patient drug *X* to treat problem *Y*").

This work examines the problem of tracking a given medication over the course of the patient's medical history, thereby reconstructing an accurate picture of what drugs (and at what dosages) the patient was on at any given time. This information is intuitively presented as a graphical timeline. Such a system would enable clinicians to more readily understand a patient's drug/medication history and prevent possible complications by automatically creating a centralized list of drugs on a timeline, as opposed to inspecting all the reports individually. Ultimately, tracking and summarizing the medication information automatically would allow physicians to avoid the time-consuming and possibly inaccurate process of manually assessing a patient's medical history.

Background

There is earlier work focused on extracting specific entities from medical reports. For example, Friedman et al. have worked on extracting relevant findings from pathology reports [10] and pneumonia information from radiology reports [11]. Medical lexicons have been used in automated problem list generation [12] and extracting noun phrases [13]. In *EDGAR* [14], information about drugs and genes relevant to cancer are extracted from the biomedical literature. More specifically, dosage identification has also been previously studied. For instance, Evans et al. [15]

worked on this problem, using an approach based on UMLS (Unified Medical Language System) [16].

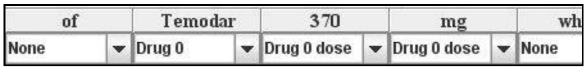
Unfortunately, tracking drug use and dosage over time remains an open problem, aspects of which are related to the well-known issue of co-reference resolution ([17,18]): references to the same drug in different reports need to be matched. In this paper, an attempt is made at examining the issues surrounding this challenge.

Methods

The proposed approach combines predefined rules (regular expressions) with a classifier. The steps are listed briefly as follows, and elaborated upon subsequently:

1. Each document's report text is broken down into sentences. Drug names are found by looking up every word token in a sentence against a publicly available FDA (Food and Drug Administration) database (Drugs@FDA).
2. Once a drug name is found, regular expressions are used to discover its associated dosage.
3. A second set of regular expressions is employed to calculate temporal features for each identified drug. These features are fed to a classifier that provides an estimate of the "state" of the drug at the time of the report.
4. Once all reports have been analyzed, a timeline is created with the evolution of the drug states.

Data and gold standard creation**.** As an initial study, a corpus of 923 documents from 11 different patients was extracted from the hospital information system of the University of California, Los Angeles. The selected patients were randomly chosen from a cross-section of records for individuals with a spectrum of medical problems (*e.g.,* various cancers, urological and neurological problems, diabetes, etc.). The documents encompassed all records in the EMR, including radiology, pathology, and admission/discharge summaries (*i.e.,* no reports were filtered out). The reports were manually annotated to create a gold standard for evaluation. A Java-based tool was designed for the simplification of this task. This tool automatically parses the text into sentences, finds drug names by comparing the words to the FDA drug list, and then uses regular expressions to identify dosages. The dosages are further characterized by amount (*e.g.,* 200 mg) and frequency (*e.g.,* b.i.d.; daily; etc.). The annotation tool presents this information to the user for validation and correction. (Fig. 1).



of	Temodar	370	mg	wh
None ▾	Drug 0 ▾	Drug 0 dose ▾	Drug 0 dose ▾	None

Figure 1. Java-based annotation tool. Word labels are assigned to generate a gold standard and for training data.

Once all the sentences containing drug names are annotated, the user is asked to define each drug reference as being ON or OFF (i.e., being taken by the patient or not)

The annotation tool currently does not support annotating dosages outside the current sentence (*e.g.,* "*The patient is to take acetaminophen. A dose of 500 mg four times a day is suggested.*"). However, as evaluating the dosage extraction is not the main goal of the current study, and as we aim to speed the tagging process across a large number of reports, we made the simplifying assumption that dosages related to a given drug are in the same sentence as the drug.

Sentence parsing**.** To break the report down into sentences, the Java package, *LingPipe*, (http://alias-i.com/lingpipe/) was used. LingPipe provides a heuristic sentence model whose rules are based on biomedical research abstracts found in MEDLINE [19]; as such, this program is well suited to the boundary detection task in the medical reports. The underlying performance of LingPipe was not evaluated.

Drug identification. For each sentence, all words and all word pairs are compared against a list of drugs and their corresponding active ingredients in a database. HSQLDB, a structured query language (SQL) relational database engine written in Java, was used to store the contents of the FDA drug list files. Three aspects of this data list were saved: 1) a unique identifier for each drug, represented by an integer; 2) the medication brand name; and 3) the active ingredient of the drug.

Dosage extraction. Given our assumption about the concurrent appearance of drug names and dosages in the same sentence, the system uses three different, expert-defined regular expressions (dependant on the type of medication) to look for dosage patterns (Table 1). If any of the three regular expressions are found, the matching portion of text is tagged as "dosage". Once a dosage is found, 25 other regular expressions are used to find the frequency with which the drug is to be taken. These regular expressions take into account the common abbreviations used by medical professionals in prescribing medications (sample in Table 2). As with dosage, for every frequency match, the corresponding text is assigned the label "frequency." The performance of this identification algorithm was evaluated against the gold standard.

<table>
<tr><td>"((one)|(two)|(three)|(four)|(five)|(six)|(seven)|(eight)|(nine)|(ten)|(to)|([0-9]*\\.?,?/?[0-9]+))+ " #drug# " ((tablets)|(tablet)|(tabs)|(tab))"</td></tr>
<tr><td>"((one)|(two)|(three)|(four)|(five)|(six)|(seven)|(eight)|(nine)|(ten)|(to)|([0-9]*\\.?,?/?[0-9]+))+\\s?((mg)|(mg/cc)|(gm)|(gram)|(mcg)|(inh)|(puffs)|(puff)|(mEq)|(ml)|(ML)|(spray)|(sprays)|(drip)|(tablets)|(tablet)|(tabs)|(tab)|(capaules)|(capsule)|(caps)|(tinc)|(tincture))(of)?\\s?" # drug</td></tr>
<tr><td>drug # "\\s?((infusion)|(MDI)|(CD)|(to)|(up to))?,?" + "(\\s?((one(\\s?to)?)|(two(\\s?to)?)|(three(\\s?to)?)|(four(\\s?to)?)|(five(\\s?to)?)|(six(\\s?to)?)|(seven(\\s?to)?)|(eight(\\s?to)?)|(nine(\\s?to)?)|(ten(\\s?to)?)|([0-9]*\\.?,?/?[0-9]+))+\\s?((mg)|(gm)|(gram)|(mcg)|(inh)|(puffs)|(puff)|(mEq)|(ml)|(ML)|(sprays)|(spray)|(drip)|(tablets)|(tablet)|(tabs)|(tab)|(capsules)|(capsule)|(caps)|(tinc)|(tincture)))"</td></tr>
</table>

Table 1. Simple regular expressions for dosage extraction. The drug name under analysis is represented by *drug*. The operator # represents concatenation.

<table>
<tr><td>drugAndDosage + "\\s? " + "p?\\.?o?\\.?\\s?q\\.?[0-9]+\\.?h?\\.?\\s?(p\\.?r\\.?n\\.?)?"</td></tr>
<tr><td>drugAndDosage + "\\s? " + "p?\\.?o?\\.?\\s?as needed"</td></tr>
<tr><td>drugAndDosage + "\\s? " + "p?\\.?o?\\.?\\s?q\\.?i\\.?d\\.?\\s?,?\\s?(p\\.?r\\.?n\\.?)?"</td></tr>
</table>

Table 2. Three sample regular expressions for extracting the frequency of drug use; 25 were used in total. Here, *drugAndDosage* represents a string containing the drug and its dosage.

Drug state classification. Following the drug identification step, two expert-defined regular expressions are compared against the sentences containing the detected drug names. The first one carries sub-expressions associated to starting a new medication or continuing a previous one (ON state), while the second includes sub-expressions associated to terminating the use of a drug or irrelevant references such as allergies, refusals or past medications (OFF state). Examples of these expressions are shown in Table 3.

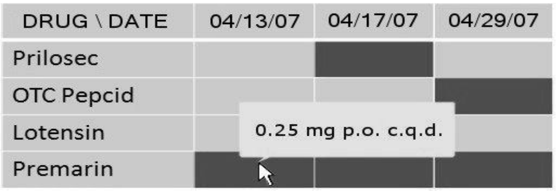
<table>
<tr><td>State ON:
((take "+drug+")|(medication)|(current)|(will [^(not)])|(added"+drug+")|(restarted)|((current)|(discharge)|(medication)|(medications?at home)|(continu)|(keep)|(presently)|(stay)|(taper)|(cut down)|(change)|(increase)|(decrease)|(turn(ed)?((down)|(up)))|(wean)|(as needed)|...</td></tr>
<tr><td>State OFF:
((allerg)|(no "+drug+" was needed)|(complains)|(toler)|(refuse)|(never)|(not)|(response)|(denies)|(deny)|(avoids?)|(wish)|(trial)|(without)|(was withheld)|(discontinue)|(underwent)|(seemed to work)|(years ago)| |(stop)|(will not)|(withheld)|(taken off)|(held)|(titra)...</td></tr>
</table>

Table 3. Regular expressions for calculating the features that the classifier uses to determine the state of a drug (only the first four lines are shown for each expression).

One simple feature is defined for each of the two regular expressions: feature *i* (where i={1 (ON) or 2 (OFF)}) for drug *d* in report *r* represents the number of times that the regular expression *i* has been matched by a sentence containing drug *d* in report *r*. This simple set of two discrete features is then fed to a support vector machine that determines the drug state (*i.e.,* ON, OFF) at the time of each report. The simplicity of the feature set enhances the generalization ability of the classifier: pilot experiments showed that this approach outperformed our efforts to include more complicated strategies based on applying feature selection [20-21] or principal component analysis (PCA) [22] to a larger set of features.

The evaluation of this classification task was performed in a "leave one patient out" manner (*i.e., n*-fold cross-validation across 11 sets); the classifier is trained using all but one of the patients, then evaluated on the remaining patient. The evaluation process is repeated for every patient and the individual results aggregated. Therefore, all the available data is used for the evaluation, while ensuring that the classifier has not been trained with testing data.

Timeline. The classifier output for all the patient's medications is displayed on a timeline in a web-based graphical interface (Fig. 2). This way, a physician can quickly and easily visualize and assess a patient's medication history. The application uses a color code for the state of the drugs, and when the user places the mouse over a particular medication on the timeline, the dosage and use frequency information are displayed.



DRUG \ DATE	04/13/07	04/17/07	04/29/07
Prilosec		■	
OTC Pepcid			■
Lotensin			
Premarin	■	■	■

0.25 mg p.o. c.q.d.

Figure 2. Timeline visualization with the results of the drug tracking process. A dark cell means that a drug is ON.

Results

The results of the dosage extraction information are displayed in Table 4. Although the sensitivity is not exceptionally high, the number of false positives is minimal.

Dosage (*e.g.,*200 mg) 2149 occurrences	Detected (%)	2015 (93.75%)
	FPs (per occurrence)	46 (0.02)
Frequency (*e.g.,* b.i.d.) 1985 occurrences	Detected (%)	1172 (84.21%)
	FPs (per occurrence)	132 (0.07)

Table 4: Sensitivity and number of false positives (FP) per evaluated drug for the dosage extraction algorithm.

For the task of drug state classificaton, we plot the receiver operating characteristic (ROC) curve by applying different weights to the two types of error (ON→OFF, OFF→ON) in the training stage (fig. 3). The area under the curve, which measures the goodness of the classification algorithm, is A_z=*0.856.* If the point of the curve corresponds to equal weights for both types of errors is selected as the operating point, the classifier presents approximately equal values for its precision and its recall: *87%* (see confusion matrix in Table 5 and derived values in Table 6).

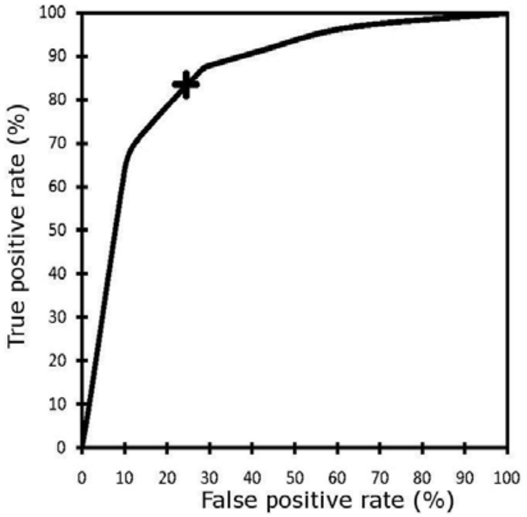


Figure 3. ROC curve for the drug state classifier. The area under the curve is A_z=*0.856*. The operating point has been marked with a cross.

Output Truth	OFF	ON
OFF	586	236
ON	240	1693

Table 5: Confusion matrix at selected operating point.

Positive predictive value (precision)	87.77%
True positive rate (recall)	87.58%
False positive rate	28.71%

Table 6: Performance metrics derived from the confusion matrix in Table 5.

Discussion

A system designed to track a patient's medications and prescribed dosages over a series of medical documents is presented. To establish a gold standard, a Java-based annotation tool was developed to quickly facilitate manual medication identification and classification. The output of the classification is presented on a web-based timeline.

Through the manual evaluation process, it was evident that several medications were not realized by the system. In some cases, some experimental drug names (*e.g.,* chemotherapies) were not in the FDA list. Also, while minimal, typographical errors in medication names (which occurred sparingly in our evaluation) were not handled by our system. The fact that not all drug names are identified as medications does not bias the experiments significantly: the performance of the algorithm should not be affected by an increase in total medications, so the incomplete medication list should not present a problem. Because the main objective of this study was to track the identified medications, the accuracy in the labeling of the medications themselves as true medications was not evaluated. Clearly, if a clinical system is to be created, it should be based on a more comprehensive database than the FDA drug list and contain a set of common misspellings or otherwise handle typographical errors in the drug matching process.

In this study the dosage and frequency extraction was not perfect, particularly because of the simplifying assumption that drug dosages would be in the same sentence as the drug name. While this was the case for the majority of dosage occurances in our sample reports, in some situations the drug dosage was found in other ensuing parts of the text. Based on this rudimentary method, the dosage extraction performance is acceptable, but it could be improved by replacing it with a medical natural language processing (NLP) engine. However, accurately extracting the dosage and frequency is not the main purpose of this study.

The dominant focus of this work is the drug state classification. Selecting features for this categorization is a difficult problem. In pilot experiments we found that a very reduced number of features generalized much better than other complicated approaches, such as feature selection and PCA. The achieved values for precision and recall (*87%*) are promising, given the difficult nature of the problem; for coreference resolution problems, results above *80%* are rarely reported [17,18].

As a relatively low number of reports were used from a small patient test set, and considering the high degree of language variability within free-text medical documents, the number of features should be kept low. This is important in order to not affect the ability of our algorithm to generalize larger test sets from multiple sources.

Conclusion

A first step into the problem of tracking drugs along a patient's medical record has been taken in this paper. The problem is a very difficult task due to the enormous complexity of free-text and language nuances.

Future work must first be directed towards enlarging the database of annotations. Even if 923 reports represent a reasonable amount, the fact that they belong to only 11 patients from the same hospital introduces bias to this work. Hence, the testbed is being extended to ensure that a wider variety and representative set of documents are used in the evaluation process.

When many more annotated reports are available, it will also be interesting to apply machine learning techniques to let the system select the most useful features from the free text training data, rather than using a hybrid approach where the importance given to human knowledge is large. Finally, when the drug state classifier has been optimized, it will be combined with state-of-the-art dosage and frequency extraction tools (rather than the prototype used here) and integrated with the timeline visualization.

References

1. Cios K, Moore G. Uniqueness of medical data mining. Artificial Intel Medicine. 2002;26:1-24.
2. Bush, R.W. Reducing waste in US health care systems. J Am Med Assoc. 297:871-874. 2007
3. Kalmeijer MD, Holtzer W, et al. Implementation of a computerized physician medication order entry system at the Academic Medical Centre in Amsterdam. J Pharmacy World and Science. 2003;25:88-93.
4. Greenberg RN. Overview of patient compliance with medication dosing: a literature review. Clin Ther. 1984;6:592-599.
5. Kim MI, Johnson KB. Personal health records: evaluation of functionality and utility. J Am Med Inform Assoc. 2002;9:171-180.
6. Miller RH, Sim I. Physicians' use of electronic medical records: barriers and solutions. Health Affairs. 2004; 23:116-126.
7. Stead WW, Kelly BJ, Kolodner RM. Achievable steps toward building a national health information infrastructure in the United States. J Am Med Inform Assoc. 2005; 12:113-120.
8. Bates DW, Ebell M, Gotlieb E, Zapp J, Mullins, HC. A proposal for electronic medical records in U.S. primary care. J Am Med Inform Assoc. 2003; 10:1-10.
9. Wang SJ, Middleton B, et al. A cost-benefit analysis of electronic medical records in primary care. Am J Med, 2003; 114:397-403.
10. Xu H, Anderson K, Grann VR, Friedman C. Facilitating cancer research using natural language processing of pathology reports. In Proc of the 11th World Congress on Health (Medical) Informatics. 2004;:565-572.
11. Mendonca EA, Haas J, Shagina L, Larson E, Friedman C. Extracting information on pneumonia in infants using natural language processing of radiology reports. J Biomed Inf. 2005:38:314-321.
12. Meystre S, Haug P. Automation of a problem list using natural language processing. BMC Med Informatics and Decision Making. 2005; 5:30.
13. Huang Y, Lowe HJ, et al. Improved identification of noun phrases in clinical radiology reports using a high-performance statistical natural language parser. J Am Med Inform Assoc. 2005;12:275-285.
14. Rindflesch TC, Tanabe L, et al. EDGAR: extraction of drugs, genes and relations from the biomedical literature. Pac Symp Biocomput. 2000; 5:514-525.
15. Evans DA, Brownlow ND, et al. Automating concept identification in the electronic medical record: an experiment in extracting dosage information. Proc AMIA Annu Fall Symp. 1996; pp. 388-92.
16. Humphreys BL, Lindberg DAB. The UMLS project: Making the conceptual connection between users and the information they need. The Unified Medical Language System: current research and development. Bull. Med. Libr. Assoc. 1993; 81:170-177.
17. Soon WM, Ng HT, Lim DCY. A Machine Learning Approach to Coreference Resolution of Noun Phrases. Assoc Comput Ling. 2001; 27(4):521-544.
18. Ng V, Cardie C. Improving machine learning approaches to coreference resolution. Proc. Annu Meet Assoc Comput Ling. 2002; pp. 104-111.
19. Greenhalgh T. How to read a paper: The MedLine database. BMJ 1997; 315:180-183.
20. Pudil P, Novovičová J, Kittler J. Floating search methods in feature selection. Pattern Recognition Letters. 1994; 15:1119-1125.
21. Jain A, Zongker D. Feature selection: evaluation, application, and small sample performance. 1997; 19:153-158.
22. Jollife IT. Principal Component Analysis. 2nd edition. 2002. Springer.