

Drug Prescription Validation based on Web and Prescription Database Mining for Medication Error Reduction

Wilson Wong¹ and David Glance

*Centre for Software Practice
University of Western Australia
Crawley WA 6009*

Abstract

Objective: Data show that medication errors are becoming increasingly frequent, with most occurring at the prescription stage. Some of the common mistakes include unclear or incorrect doses or frequency, specifying the wrong medicine, and omitting or delaying medicines. Many challenges have continued to prevent existing solutions such as the Computerised Physician Order Entry from being replicated to the outpatient setting, especially small and rural health facilities. This paper look at three technical challenges, namely, the problem of knowledge maintenance, complex interface that disrupts work flow, and intolerance to noisy prescription text.

Methods and Materials: This paper presents a prescription text validation technique based on three sources of background knowledge, namely, the World Wide Web, the Australian Medical Terminology, and past prescribing data on the MMEx electronic health record system, for reducing two common prescribing errors of incorrect dosage/frequency and drug name confusion. **Results:** Our evaluation using 190 actual prescriptions containing 30 medication errors and 30 spelling mistakes showed an accuracy range between 95.26% and 96.32%. **Conclusion:** The technique validates prescription text via a simple interface which is reminiscence of the conventional prescription pads. The technique does not require complex knowledge bases and the accuracy of the technique degrades gracefully in the face of spelling errors. **Keywords:** Prescription text; Medication Error; Drug name confusion; Prescription validation; Prescription bracketing; Prescription habit analysis

¹ Corresponding author. Tel: +61 864882839, Fax: +61 64881089

1. Introduction

Medication incidents refer to any problems related to medication. Preventable incidents that may lead to inappropriate medication use, regardless of whether any injury occurred is referred to as *medication errors* [1]. Medication errors can occur at any stage during the medication process, which includes prescribing, dispensing, administration and monitoring. A small proportion of incidents that do result in injuries to patients are known as *adverse drug events* (ADEs). ADEs that result from unintended response to drugs at normal doses are called *adverse drug reactions* (ADRs), while those that take place due to medication errors are referred to simply as preventable. Figure 1, adapted from [2], summarises the types of medication incident.

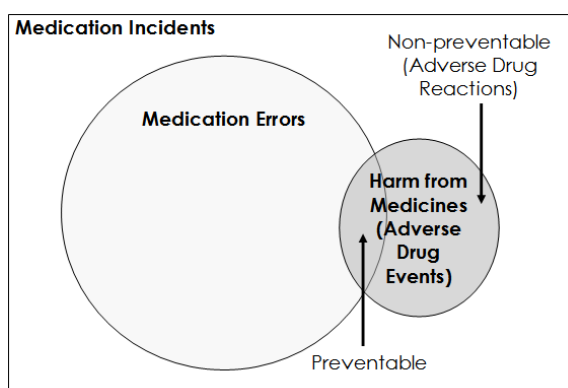


Fig. 1. The types of medication incident.

Medication errors have reached epidemic proportions in developed countries. Medication errors are estimated to be the leading cause of medical mistakes in US hospitals, accounting for nearly 20% of all errors according to a report [3] by the Institute of Medicine of the US National Academies. The report further stated that medication errors result in over 7,000 deaths each year, and argued that prescription errors are rising dramatically. The UK National Patient Safety Agency (NPSA) also noted an increase in medication incidents from 36,335 in 2005 to 86,085 in 2007 [2]. In Australia, about 10% of the 17.5 million people who make 95 million visits to their general practitioners annually report experiencing ADEs [1,4].

Many studies have found that most medication errors occur during the prescribing stage [5–7]. In Australia, the rate of prescribing errors were found to be as high 115 per 100 high risk patients [1]. The majority of fatal and serious injuries from medication errors (up to 71% in one report by the UK NPSA [2]) are due to unclear or incorrect dose or frequency, wrong medicine, and omitted or delayed medicines [8]. Regulatory agencies, the pharmaceutical industry, information providers, safety and quality organisations, professional bodies, health professionals and patients all have a part to play in reducing prescribing error and its role in ADEs [9]. Information and communication technologies, however, have been advocated as one of the most effective means of doing so [10]. One solution, in particular, that is slowly demonstrating convincing results is electronic prescribing using *Computerised Physician Order Entry* (CPOE) systems [11].

CPOE systems with decision support capabilities have been reported to be successful in reducing medication errors in inpatient settings [11,10,12,13]. However, attributes synonymous with complex CPOE systems such as the requirement for expert maintenance of knowledge bases and the major disruption to workflow contribute to the slow adoption of prescribing systems with decision support capabilities in outpatient settings. Therefore, the long term goal of this research is to introduce and deliver feasible prescribing systems with decision support capabilities to the private and public health services throughout Western Australia as part of the MMEx electronic health record system (<http://www.gsmhn.com.au>) offered by the University of Western Australia (UWA) Centre for Software Practice (CSP).

In this paper, we report the results of a project that looked at one particular aspect of our long term goal, which is to develop and evaluate a user-friendly interface for *prescription text validation* to promote safer prescribing without the three problems, namely, (1) the tedious process of knowledge maintenance, (2) complex interfaces that disrupt workflows, and (3) the lack of robustness to withstand noisy prescription text. The proposed validation technique addresses two common prescribing errors, namely, incorrect dosage/frequency and drug name confusion, through the analysis of medical knowledge from the World Wide Web, the Australia Medical Terminology (AMT), and past prescribing data from the MMEx system. Section 2 discusses related work on CPOE and the MMEx electronic health record systems. Section 3 describes the proposed technique for validating prescription texts. The evaluation results of the proposed technique are provided in Section 4. We conclude this paper in Section 6.

2. Related work

This section provides an overview of CPOE, and its application in inpatient and outpatient settings. In addition, we look at several techniques for processing medication information which can be used to implement decision support capabilities or to construct and maintain clinical databases. A brief summary of the MMEx system, in which the proposed validation technique is based, will also be provided.

2.1. Computerised physician order entry

CPOE refers to a variety of computer-based systems for entering medical orders (e.g. laboratory, radiology, or medication). Besides addressing the obvious problem of illegible handwriting, CPOE systems can range from systems that only provide a list of possible medications that physicians can choose from, to systems providing varying levels of decision support, including checks of drug names, doses, routes, frequency, patient allergies, and drug interactions. All these checks lead to alerts and reminders given to the physician in case problems are detected, thereby helping to ensure that medications are prescribed appropriately. Full-fledged CPOE systems deployed in large medical centers have been shown to reduce medication errors at varying degree of success in inpatient settings [11,10,12,13].

The more advanced CPOE systems, which incorporate decision support capabilities, typically employ some form of knowledge base and inference engine, or use machine learning techniques to acquire and recognise patterns in clinical information. Systems

such as HELP [14] at the LDS Hospital in Utah, and the WizOrder system [15] at the Vanderbilt University Medical Center, which were conceived during the 1980s to the 1990s, rely heavily on human experts to create and maintain rules and knowledge for the systems to remain relevant. Recognising the maintenance bottleneck, systems such as WizOrder have begun to practice collaborative maintenance by providing distributed tools to the various stakeholders to contribute to the continuous evolution of the knowledge bases [16]. Recently, researchers are beginning to acknowledge the role of resources on the World Wide Web in enhancing clinical decision support. [17], in their study, noted the importance of online dictionaries, and found that different dictionaries are capable of addressing the different needs of clinical decision support. The need for a more comprehensive solution to the bottleneck of knowledge maintenance has been taken a step further through the use of Web 2.0 technologies. [18] examined several efforts for sharing and collaborating on decision support content such as Clinfowiki, Partners HealthCare eRooms, and Epic Systems Corporation's Community Library.

These advanced CPOE systems are built to work in rich information environments consisting of hospital-based electronic medical record and other manually-maintained knowledge bases that cannot be easily replicated in outpatient settings. Unlike large medical centers, small private health facilities and rural hospitals do not have the necessary expertise (e.g. informaticians, expert clinicians), infrastructures and huge budget to develop or purchase, customise and roll out these advanced systems. In outpatient settings where CPOE systems are not available, technology is mainly used for electronic billing, electronic appointment scheduling, electronic medical/drug information and e-mail communications to patients [19]. However, due to government-imposed reforms (such as the US Medicare reform [20]) and the increasing demand for computerised solution to medication error outside of large medical centers, prescribing systems with decision support capabilities in outpatient settings is slowly gaining popularity [21,22].

2.2. Techniques for medication information processing

Evans et al. [23] conducted one of the earliest study on the extraction of drug and dosage information from unstructured clinical information using existing technology and resources. In their study, the authors employed natural language processing tools from a proprietary system called CLARIT, and the 1996 edition of the Unified Medical Language System (UMLS) to create an extraction module for processing clinical notes.

Shu et al. [24] presented an interactive Java-based application for encoding events in clinical data from intensive care units with the most appropriate clinical concepts in UMLS. The authors identified spelling errors, abbreviation ambiguities and the absence of UMLS code for certain events as some of the common challenges that arise during the encoding process. To address these problems, the authors introduced into the application a spell-checker and a personalised dictionary that allows users to define uncommon and new abbreviations.

Shah & Martinez [25] developed a technique to extract numerical information from unstructured dosage instructions. The technique first normalises words and phrases in the dosage instructions using a dictionary. The standardised instructions are then split into parts using words such as "*and*", "*or*" and "*max*". The parts of the instructions are analyzed separately using patterns and dictionaries. Structured numerical information

such as dose quantity, frequency, units and duration are identified and presented in the E2B format. The technique was implemented as a Visual Basic program for use with data kept in Microsoft Excel files.

More recently, Gold et al. [26] presented a technique for extracting medication information including drug names appearing with or without dosage information, and even misspelt drug names. The technique employs manually-defined rules, implemented as a set of regular expressions, and a user-configurable drug lexicon. In this technique, a non-drug-name phrase is considered as a misspelt drug name or a drug name not present in the lexicon if it is surrounded by adequate dosage information. The authors tested their technique using 26 discharge summaries. The authors reported that their technique achieved a precision of 94% and recall of 83% in the extraction of medication information.

2.3. MME_x electronic health record system

UWA CSP, within which this research work will be based, has been working with the Department of Health Western Australia (DoHWA), Aboriginal Community Controlled Health Services and private practice to implement electronic health record capabilities. The platform that this is being delivered on is MME_x (<http://www.gsmhn.com.au>), which currently supports approximately 6,500 health professionals with around 300,000 patient records and transmitting 10,000 secure electronic messages per month. MME_x provides a full, shareable electronic health record for patients along with functionality and modes of collaboration to support all health professional types. Through collaboration with DoHWA, MME_x will become a key infrastructure component of the DoHWA's e-Health platform, especially in rural and remote settings. The growing number of health professionals relying on MME_x for managing patient records increases the importance of making the information accessible in its structured and context-dependent form. This is especially the case when the information is required for decision support and other intelligent functions to assist collaborative teams of health professionals in delivering primary and acute care to patients. For this reason, UWA CSP has been putting together a long term plan to incorporate CPOE together with decision support capabilities into MME_x. This paper presents the results from an initial attempt to introduce prescription validation to promote safer prescribing amongst general practitioners in remote areas of Western Australia (WA).

3. Prescription text validation

This paper presents a technique for validating multiple lines of prescription text to detect and prevent potential medication errors caused by incorrect dosage/frequency and drug name confusion. The proposed validation technique comprises two processes. The first process, known as *prescription bracketing*, structures free-text prescriptions into three components (i.e. drug name, dosage, frequency). The second process, called *prescribing habit analysis*, analyzes the structured prescriptions and alerts the physicians of any potential medication errors. An override mechanism is also incorporated into the technique to allow physicians to assign trust and accept uncommon prescriptions. Figure 2 and Algorithm 1 provides an overview of the technique. The details about

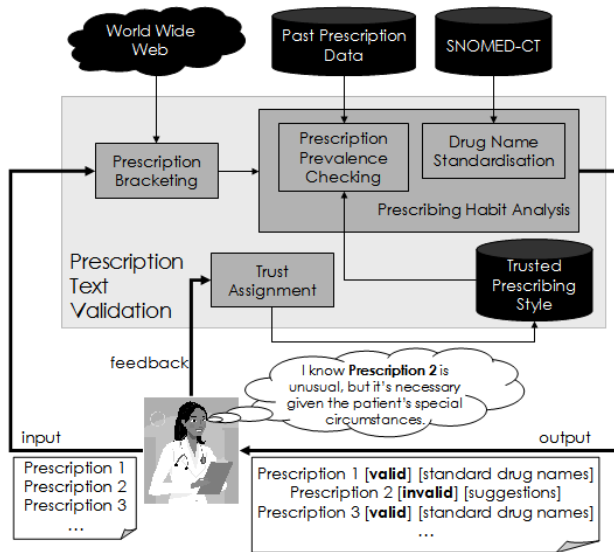


Fig. 2. An overview of the proposed validation technique.

prescription bracketing and prescribing habit analysis are described in Sections 3.1 and 3.2, respectively.

The three characteristics that sets the proposed technique apart from existing attempts at prescription validation are:

- The ability to automatically harness raw prescribing data of heterogeneous nature from both the MMEx system and the World Wide Web to compensate for the absence of manual knowledge maintenance;
 - A simple prescribing interface comprising of only a text field that is reminiscent of past prescribing practices rather than complex CPOE systems; and
 - The robustness to withstand potentially noisy (e.g. spelling errors) prescription texts.
- Conventional solutions in existing CPOE technologies disrupt normal work flow and suffer from the overhead of knowledge maintenance. The inability to deal with spelling errors in prescription text may also be present. For instance, a recently patented technique by [27] relies on manually-defined patterns for parsing prescription text and drug databases for performing validation. This system simply alerts the physician of invalid or incorrect information when no database match is found. No means is provided to cope with spelling errors and suggest alternatives.

In the proposed technique, the physician enters traditional prescription text into a user interface, as shown in Figure 3(a), that is analogous to a prescription pad. The text is then automatically analyzed and structured into a standard format, resulting in a prescribing process that is more similar to past prescribing practices than existing CPOE systems. We refer to this structuring process as prescription bracketing. This process uses drug names and related information available on the World Wide Web to dynamically determine the appropriate conditions for breaking down the prescription text into its components comprising drug names, dosage and frequency. Figure 3(a) provides the bracketing output for the input “*Zyban 150mg Tablet one twice a day*”. Algorithm 2 provides an overview of the bracketing process.

Algorithm 1 `validate(prescriptionlines)`

```
1: break the multiple lines of prescription strings prescriptionlines into set S.
2: for each  $X \in S$  do
3:    $\{drugname, dose, freq, misc\} := bracket(X)$ 
4:    $E := check(drugname, dose, freq, misc)$ 
5:    $G := standardise(drugname)$ 
6:   display the bracketing and analysis output in  $\{drugname, dose, freq, misc\}$ ,  $E$  and
      $G$  to the users as shown in Figures 3(a) and 3(c).
7: end for
```

The structured prescription data provided by the bracketing process is then analyzed and validated. Drug information on SNOMED-CT, and prescribing habits deduced from non-identifiable data on MMEx is used to catch any anomalous prescriptions (e.g. incorrect drug names, wrong dosage) in the structured input. The prescribing habit analysis process analyzes past prescribing data on MMEx to determine the validity of the whole or parts of the structured input prescription. Fuzzy string comparison algorithms are employed during the analysis process to ensure that misspelt drug names or prescriptions will have minimal impact on the performance of the technique. The details about alternative drug names and the validity of the input prescription are consolidated and returned to the physicians. Figure 3(c) provides a screenshot of the output of the prescribing habit analysis process.

In the case of detecting an invalid prescription, our validation technique alerts the physician about the anomalous nature of that particular prescribed medication and provide suggestions. Figure 3(b) shows an alert provided to the physician for the input prescription “*Zyban 150mg Tablet two every two hours*”. Note that from past prescriptions, the recommended dose for the antidepressant and smoking cessation aid Zyban is once a day. The input prescription of two tablets taken every two hours definitely constitutes an overdose, and our technique was able to detect that. The physician then has the choice of either selecting from the list of suggested prescriptions, or insisting on accepting and proceeding with the medication as prescribed.

3.1. Prescription bracketing

The prescription bracketing process analyzes and structures the free-text input string X into drug name, dosage and frequency data, here after known as the *components* of the prescription. Each $X = x_1x_2\dots x_m$ is made up of a sequence of words that can be broken down into contiguous substrings to represent the different prescription components. Since the input is free text, there are no a priori demarcation lines to assist in identifying the components of X . Moreover, attempts to separate them using typical natural language processing (NLP) tools (e.g. text segmentation, sentence parsing) is infeasible since the construction of X do not follow grammar rules and sentence boundary conventions. To address these problems, we devised a new statistical based technique which combines *pointwise mutual information* (ρ) with a new heuristically-derived measure known as *balance* (β).

The `bracket` process in Algorithm 2 summarises our prescription bracketing technique. The `bracket` process accepts a single prescription string X of any length and returns a

prescription validation

prescription text:

[proceed](#)

[\[+\] analysis results](#)

[\[+\] bracketing results](#)

processing X="Zyban 150mg Tablet one twice a day"

Zyban (12600000) + 150mg Tablet one twice a day (0) = (0)[β:8E-08][δ:0][ρ:100000000]

Zyban 150mg (109000) + Tablet one twice a day (3) = (0)[β:3.67E-05][δ:1][ρ:100000000]

Zyban 150mg Tablet (128) + one twice a day (14500) = (0)[β:0.00881819][δ:1][ρ:100000000]

Zyban 150mg Tablet one (0) + twice a day (15400000) = (0)[β:6E-08][δ:0][ρ:100000000]

Zyban 150mg Tablet one twice (0) + a day (771000000) = (0)[β:0][δ:0][ρ:100000000]

(a_p=""), (ρ_k=100000000), (a_β="Zyban 150mg Tablet")(β_j=0.00881819)

condition 2 (notstable && paircnt > 1) was met

processing X="one twice a day"

two pairs remain [one twice a day][stability:-31.754091611188][stable? FALSE]

one (11900000000) + twice a day (15400000) = (14500)[β:0.00129245][δ:1][ρ:-8.3364547]

one twice (613000) + a day (771000000) = (14500)[β:0.00079444][δ:1][ρ:0.26253167]

(a_p="one"), (ρ_k=-8.3364547), (a_β="one")(β_j=0.00129245)

condition 1 (notstable && p <= -4.8) was met

processing X="twice a day"

twice (556000000) + a day (771000000) = (1540000)[β:0.60040253][δ:1][ρ:0.49021131]

(a_p="twice"), (ρ_k=0.49021131), (a_β="twice")(β_j=0.60040253)

condition 3 was met

Zyban 150mg Tablet[one][twice a day][

(a) The output of prescription bracketing for the string "Zyban 150mg Tablet one twice a day".

prescription validation

prescription text:

[proceed](#)

[\[+\] analysis results](#)

[drug name: zyban 150mg tablet] [dose: two] [freq: every two hours]

your prescription is NOT VALID!

	did you mean?
accept the medication as prescribed	<input type="checkbox"/>
[zyban tablet][one][] (prescribed 12 times)	<input type="checkbox"/>
[zyban 150mg tablet][one][] (prescribed 7 times)	<input type="checkbox"/>
[zyban 150mg tablet][one][daily] (prescribed 6 times)	<input type="checkbox"/>
[zyban 150mg tablet][][in the morning] (prescribed 1 times)	<input type="checkbox"/>

details of drugs in your prescription

- [Zyban SR][bupropion hydrochloride 150 mg][modified release tablet][trade product][prob:100%]

[\[+\] bracketing results](#)

(b) The output of prescribing habit analysis for the invalid input "Zyban 150mg Tablet two every two hours".

prescription validation

prescription text:

[proceed](#)

[\[+\] analysis results](#)

[drug name: zyban 150mg tablet] [dose: one] [freq: twice a day]

your prescription is **VALID** based on previous similar prescriptions, with a min. confidence of **100% (0.88)**

[zyban 150mg tablet][one][twice a day] (100%)(prescribed 19 times)

[zyban 150mg tablet][one][] (100%)(prescribed 7 times)

[zyban 150mg tablet][][twice a day] (100%)(prescribed 1 times)

details of drugs in your prescription

- [Zyban SR][bupropion hydrochloride 150 mg][modified release tablet][trade product][prob:100%]

[\[+\] bracketing results](#)

(c) The output of prescribing habit analysis for the string "Zyban 150mg Tablet one twice a day".

Fig. 3. System outputs

set of structured prescription components {drugname, dosage, freq, misc}. The bracket process uses the split process described in Algorithm 3 to iteratively break X into

Algorithm 2 `bracket(X)`

```
1: initialise drugname, dose, freq, misc to empty strings
2: initialise set skipword := {a, an, the, every}.
3: initialise m to the no. of words in  $X$ .
4: while  $X$  is not an empty string do
5:   bestsubstr := split( $X$ , m, skipword)
6:   if drugname is an empty string then
7:     drugname := bestsubstr
8:   else if dose is an empty string then
9:     dose := bestsubstr
10:  else if freq is an empty string then
11:    freq := bestsubstr
12:  else
13:    misc := bestsubstr
14:  end if
15:  remove the substring bestsubstr from  $X$ .
16:  set m to the no. of words in revised  $X$ .
17: end while
18: return {drugname, dose, freq, misc}
```

prescription components using the ρ and β values. We provide a brief description of the `bracket` and `split` processes. First, as long as the input string X has not been reduced to an empty string, it is subjected to the `split` process (line 4-5 of Algorithm 2) described in Algorithm 3. Second, during the `split` process, X is iteratively cut at point i to generate contiguous substrings a and b (lines 8-9 in Algorithm 3) where $1 \leq i \leq m - 1$ and m is the number of words in X . For each pair of a and b , its β_i , δ_i and ρ_i are determined (line 11 in Algorithm 3). This process of breaking down X , and finding the β_i , δ_i and ρ_i values will take place for all $m - 1$ possible pairs. Third, the β values in B are adjusted according to $\sum \delta$ (lines 17-21 in Algorithm 3). Fourth, the maximum β and the minimum ρ values, and their corresponding substrings a are determined (lines 22-25 in Algorithm 3). Fifth, the specific substring a that satisfies a certain condition (lines 32-38 in Algorithm 3) will be returned to the `bracket` process as the valid prescription component for the current iteration of 3. Last, a series of conditions determines the type (i.e. drug name, dosage, frequency) of the returned component (lines 6-14 in Algorithm 2). This component is then removed from X (line 15 in Algorithm 2). The shorter, revised X undergoes the next iteration of the `split` process again until an empty string remains.

3.1.1. Balance, β

The balance measure β is used to discover the most equally commonly used pair of contiguous substrings. The balance values are computed in two parts. First, the initial balance at i (β_i) is computed as:

$$\beta_i = \left(\sqrt{\left(\frac{n_a}{n_b} - \frac{n_b}{n_a} \right)^2} \right)^{-1} \quad (1)$$

Algorithm 3 `split($X, m, skipword$)`

```
1: initialise  $paircnt := 0$ .
2: initialise  $\sum \delta := 0$ .
3: initialise boolean variable  $stable := false$ .
4: initialise sets  $B, D$  and  $P$  to  $m - 1$  elements.
5: initialise  $bestsubstr$  to empty string.
6: for  $i := 1; i < m; i ++$  do
7:   if  $x_i \notin skipword$  then
8:      $a := x_1 \dots x_i$ .
9:      $b := x_{i+1} \dots x_m$ .
10:    set  $n_a$  and  $n_b$  to the page counts of  $a$  and  $b$ , respectively.
11:    determine  $\beta_i, \delta_i$  and  $\rho_i$  using Equation 1, 2 and 3.
12:     $\sum \delta := \sum \delta + \delta_i$ 
13:    set the  $i$ -th elements of  $B, D$  and  $P$  with  $\beta_i, \delta_i$  and  $\rho_i$ , respectively.
14:     $paircnt ++$ ;
15:   end if
16: end for
17: if  $\sum \delta > 0$  then
18:   for each  $\beta_i \in B$  do
19:      $\beta_i := \beta_i \delta_i$  and update  $B$  with adjusted  $\beta_i$ .
20:   end for
21: end if
22: find the maximum  $\beta_j \in B$ .
23:  $a_\beta := x_1 \dots x_j$ .
24: find the minimum  $\rho_k \in P$ .
25:  $a_\rho := x_1 \dots x_k$ .
26: if  $paircnt = 2$  then
27:    $\sigma := \rho_u / \rho_v$  where  $\rho_u, \rho_v \in P$  and  $\rho_u < \rho_v$ .
28:   if  $\sigma > -0.2$  then
29:      $stable := true$ 
30:   end if
31: end if
32: if not  $stable$  and  $\rho_k \leq -4.8$  then
33:    $bestsubstr := a_\rho$ 
34: else if not  $stable$  and  $paircnt > 1$  then
35:    $bestsubstr := a_\beta$ 
36: else
37:    $bestsubstr := X$ 
38: end if
39: return  $bestsubstr$ 
```

where n_a and n_b are the page counts of substrings a and b , respectively. Second, a Boolean value δ at i (δ_i) is computed for each substring pair as:

$$\delta_i = (e^{-(n_a n_b)} - 1)^2 \quad (2)$$

where $\delta_i \in [0, 1]$. During this second part, if the sum of the δ values for all pairs is more than 0, the initial β_i values are adjusted by multiplying β_i with the corresponding δ_i . The

value of β ranges between $[0, 1]$. As β_i approaches 1, the two corresponding substrings will be more equally in common use. Table 1 shows an example of how to compute the balance at i (β_i) for different substring pairs (i.e. n_a and n_b). For $X = \text{“Metoprolol Tartrate tablet one twice a day”}$, the pair *“Metoprolol Tartrate tablet”* and *“one twice a day”* that has the least gap between their page counts achieves the best (largest value of) balance at $\beta_{j=3} = 0.034$. All other β values with $\delta = 0$ in this example were revised to 0 since $\sum \delta > 0$ (line 17 of Algorithm 3). As such, the *bestsubstr* is set to $a_\beta = \text{“Metoprolol Tartrate tablet”}$ based on the condition on line 34 of Algorithm 3. Intuitively, either one or both substrings in this case are in equally common use. This, in turn, can be used as an indicator that the corresponding substrings represent coherent concepts and are semantically significant. As the gap between n_a and n_b becomes larger, the value of β_i becomes smaller. The substrings in such cases are unlikely to refer to valid prescription components.

Table 1. The value of N is 39,100,000,000. $X = \text{“twice a day”}$ in the last round is no longer split since $\rho_k > -4.8$ and $\text{paircnt} = 1$. In this case, the third condition on line 36 of Algorithm 3 applies. After the **bracket** process is complete, the initial free-text prescription is structured into $\text{drugname} = \text{“Metoprolol Tartrate tablet”}$, $\text{dosage} = \text{“one”}$ and $\text{freq} = \text{“twice a day”}$.

i	a	b	n_a	n_b	β	δ	ρ	σ
$i=1$	Metoprolol	Tartrate tablet one twice a day	3,170,000	0	0.00000032	0.00	∞	
2	Metoprolol Tartrate	tablet one twice a day	147,000	3	0.00002721	1.00	∞	
3	Metoprolol Tartrate tablet	one twice a day	510	14,500	0.03408031	1.00	∞	
4	Metoprolol Tartrate tablet one	twice a day	0	15,400,000	0.00000006	0.00	∞	
5	Metoprolol Tartrate tablet one twice	a day	0	771,000,000	0.00000000	0.00	∞	
		n_X	0					
$i=1$	one	twice a day	11,900,000,000	15,400,000	0.00129245	1.00	-8.34	-31.75
2	one twice	a day	613,000	771,000,000	0.00079444	1.00	0.26	
		n_X	14,500					
$i=1$	twice	a day	556,000,000	771,000,000	0.60040253	1.00	0.49	
		n_X	15,400,000					

3.1.2. Pointwise mutual information, ρ

Pointwise mutual information ρ is used in our technique to quantify the extent of the association between the substrings a and b . Substring pairs that are the least associated indicates a high degree of syntactic and semantic independence from one another. In our context, such pairs form relatively stable phrases that refer to semantically meaningful concepts. The pointwise mutual information at point i (ρ_i) is computed as:

$$\rho_i = \begin{cases} \log_2 \left(\frac{p_X}{p_a p_b} \right) & \text{if } p_X > 0 \wedge p_a > 0 \wedge p_b > 0 \\ \infty & \text{otherwise} \end{cases} \quad (3)$$

where $p_X = n_X/N$, $p_a = n_a/N$ and $p_b = n_b/N$. n_X is the page count for string X and N is an estimation² of the index size of the search engine from which we obtain the page counts. ρ compares the probability of observing a and b together as X with the probabilities of observing them independently. If there is a genuine association between a and b , then $\rho_i > 0$ since the joint probability p_X will be much larger than chance $p_a p_b$. Otherwise, p_X will be much less than $p_a p_b$, forcing $\rho_i < 0$. Table 1 shows how the ρ values

² Function words and numbers such as *“a”*, *“an”* and *“1”* are used to query the search engine to obtain N .

influence the bracketing decision. Based on line 32 of Algorithm 3, substrings that meet the condition regarding ρ_k are considered first. In the case of $X = \text{“one twice a day”}$, the substring $a_p = \text{“one”}$ with the least $\rho_{k=1} = -8.34$, which is also less than -4.8 and is unstable ($\sigma < -0.2$), is regarded as the *bestsubstr* for this particular `split` iteration.

3.2. Prescribing habit analysis

The prescribing habit analysis process determines whether the input prescription is valid or otherwise, and provides suggestions if necessary. This analysis process comprises a *prescription prevalence checking* process and a *drug name standardisation* process. The checking process determines the prevalence of the input prescription by comparing it with historical prescribing data in the MMEx system. The standardisation process, on the other hand, normalises the drug name from the structured input data using standard drug names from the SNOMED-CT database. Both the processes employ fuzzy string matching algorithms to find prescribing data and drug names from the databases in MMEx and the SNOMED-CT database, respectively. This use of string matching algorithms introduces robustness to the overall technique in the face of spelling errors.

3.2.1. Prescription prevalence checking

Algorithm 4 describes the `check` process for determining the prevalence of a particular prescription. This `check` process accepts an input set of prescription components $\{\text{drugname}, \text{dose}, \text{freq}, \text{misc}\}$. The checking process first uses the input component *drugname* to retrieve all related past prescriptions from an MMEx’s Structure Query Language (SQL) table called `PatientMedication`. While the table `PatientMedication` contains many fields regarding the medication process, we are only interested in the three fields about drug name, dosage and frequency. A tuple E comprising three sets E_1 , E_2 and E_3 is prepared (line 9 in Algorithm 4). All distinct past prescriptions are identified and stored in E_1 as sets (line 14 in Algorithm 4) comprising drug name, dosage and frequency. The number of times each distinct past prescription occurred in the `PatientMedication` table is recorded in E_2 (line 12 and 15 in Algorithm 4). The distinct triples of drug name, dosage and frequency in E_1 is then compared against the input components *drugname*, *dose* and *freq* (lines 19-21 of Algorithm 4) using a fuzzy string matching process called `stringcompare` based on the Wagner-Fischer implementation of edit distance [28]. Given d is the edit distance between two strings provided by the Wagner-Fischer implementation, the `stringcompare` process finds the similarity between the two strings as:

$$\chi = \left(1 - e^{-1/d}\right)^{0.5} \quad (4)$$

The results of the individual comparisons χ_{dn} , χ_{ds} and χ_{fq} are combined into a single value χ (line 22 of Algorithm 4), which is then recorded in set E_3 (line 23 of Algorithm 4). The information about the prevalence of the input prescription and its possible variations, which is stored in the tuple E is returned to the caller of the `check` process.

Due to the possibility of spelling errors, we cannot perform direct table lookups for past prescriptions containing the component *drugname* (line 8 of Algorithm 4). A *sliding window based technique* is introduced to create search queries capable of retrieving the intended past prescriptions even though the provided drug names, dosage and frequency have spelling errors. Let any of the prescription components (i.e. *drugname*, *dose*, *freq*)

Algorithm 4 `check(drugname, dose, freq, misc)`

```
1: initialise querystring to empty string.
2: initialise a tuple  $E := (E_1, E_2, E_3)$  where  $E_1 := \{e_{11} := (dn, ds, fq), \dots, e_{1z}\}$ ,  $E_2 := \{e_{21} := 0, \dots, e_{2z}\}$  and  $E_3 := \{e_{31} := 0, \dots, e_{3z}\}$ .
3: let  $drugname := y_1y_2\dots y_u$ .
4: for  $i := 1; i \leq (u - w) + 1; i ++$  do
5:   for  $v := 0; v < w; v ++$  do
6:     combine substring  $y_{i+v}$  into SQL-compatible querystring.
7:   end for
8: end for
9: query the PatientMedication table for past prescriptions (the drug name dn, dosage ds and frequency fq fields) that contain drug names similar to querystring, and assign the triples  $(dn, ds, fq)$  to set D.
10: for each  $(dn, ds, fq) \in D$  do
11:   if  $e_{1i} := (dn, ds, fq)$  already exist in  $E_1$  then
12:      $e_{2i} ++$  where  $e_{2i} \in E_2$ .
13:   else
14:      $E_1 := E_1 \cup \{(dn, ds, fq)\}$ 
15:      $e_{2i} := 1$  where  $e_{2i} \in E_2$ .
16:   end if
17: end for
18: for each distinct  $e_{1i} := (dn, ds, fq) \in E_1$  do
19:    $\chi_{dn} := stringcompare(dn, drugname)$ 
20:    $\chi_{ds} := stringcompare(ds, dose)$ 
21:    $\chi_{fq} := stringcompare(fq, freq)$ 
22:    $\chi_i := \chi_{dn}\chi_{ds}\chi_{fq}$ 
23:    $E_3 := E_3 \cup \{e_{3i} := \chi_i\}$ 
24: end for
25: return E
```

be represented as a string $Y = y_1y_2\dots y_u$. Given the window size of w , there are $(u - w) + 1$ possible substrings to be generated from Y . These $(u - w) + 1$ possible substrings are then concatenated into an SQL-compatible format for querying the `PatientMedication` table. For instance, with a window size of $w = 5$, the misspelt drug name $Y = "Akinetn"$ can be split into $(u - w) + 1 = 4$ substrings "Akin", "kineo", "ineot" and "neotn". Looking for past prescriptions using the misspelt name "Akinetn" will not yield any results. However, searching the table using a logical disjunction (e.g. A or B) of the 4 substrings will provide us with prescriptions containing the intended drug whose actual name is "Akineton".

Together, the use of the sliding window based technique and the fuzzy `stringcompare` process for searching and comparing past prescriptions with the input data improves the robustness of the overall validation technique.

3.2.2. Drug name standardisation

The drug name standardisation process is summarised in Algorithm 5. The process uses the same sliding window based technique for constructing query strings to retrieve

Algorithm 5 *standardise*(*drugname*)

```
1: initialise a tuple  $G := (G_1, G_2)$  where  $G_1 := \{g_{11} := dn_1, \dots, g_{1z}\}$  and  $G_2 := \{g_{21} := 0, \dots, g_{2z}\}$ .
2: initialise querystring to empty string.
3: let drugname :=  $y_1y_2\dots y_u$ .
4: for  $i := 1; i \leq (u - w) + 1; i ++$  do
5:   for  $v := 0; v < w; v ++$  do
6:     combine substring  $y_{i+v}$  into SQL-compatible querystring.
7:   end for
8: end for
9: query the SNOMED-CT database for standard drug names ( $dn_i$ ) similar to querystring, and store them to set  $G_1$ .
10: for each  $dn_i \in G_1$  do
11:    $\chi_{dn} := \text{stringcompare}(dn_i, \text{drugname})$ 
12:    $G_2 := G_2 \cup \{g_{2i} := \chi_{dn}\}$ 
13: end for
14: return  $G$ 
```

standard drug names from the SNOMED-CT database (lines 3-9 in Algorithm 5). The retrieved standard names are stored in set G_1 . The standard drug names are then compared against the input *drugname* using the `stringcompare` process (lines 10-13 in Algorithm 5). The string similarities are stored in set G_2 . Together, the two sets are returned as a tuple G to the caller of the `standardise` process.

4. Evaluation

For the evaluation of our prescription validation technique, we have prepared two test sets comprising 190 prescription strings each. The first set consists of actual medications prescribed by physicians using the MMEx system. We randomly introduced 30 medication errors, in the form of incorrect dosage and/or frequency, to the prescriptions in this first set. The second set contains the same prescriptions as the first set. In addition, 30 prescriptions from this set were randomly selected and introduced with 1-2 word spelling errors.

4.1. Data collection

No participants were involved for data collection. This evaluation uses only existing non-identifiable data about medications prescribed by physicians to their patients within the WA Country Health Services. The data is held in a secure database by the UWA CSP on behalf of the WA Country Health Services. WA Country Health Services and the CSP have agreed to permanently de-identify a copy of the required data for use in this research. This evaluation does not deal with any patient or physician personal information. Only non-identifiable prescription data will be used in this evaluation (i.e. sufficient protection of privacy). The use of all prescription data in this evaluation has been approved by the UWA Human Research Ethics Committee.

4.2. Results

The first part of the evaluation assessed the performance of the prescription bracketing process using the first test set. First, the different components of the 190 prescriptions were manually identified and used as the benchmark for comparison. Second, we fed the 190 lines of prescription text to our prescription bracketing process. Third, the output of the bracketing process was compared against the benchmark. Out of the 190 lines, our bracketing process was able to correctly identify the components of 180 prescriptions. In other words, the accuracy of the bracketing process in this evaluation using the above-described first test set is 94.74%. The 10 prescription strings that were incorrectly bracketed are shown in Table 2.

Table 2. The 10 prescriptions from the first test set that were incorrectly bracketed.

Full Prescription String	Drug Name	Dosage	Frequency
Champix - Combo Pack Tablet 0.5mg daily	Champix - Combo	Pack Tablet	0.5mg
Chlorsig Eye Drops two drops four times a day	Chlorsig Eye Drops	two	drops
Dilaudid-Hp Injection 20mg daily	Dilaudid-Hp Injection	20mg	daily
Efexor-Xr Capsule 300mg daily	Efexor-Xr Capsule	300mg	daily
Hydroxyprogesterone Hexanoate one regular six hourly	Hydroxyprogesterone Hexanoate one	regular	six
Sigmacort 1% Cream apply twice a day	Sigmacort 1%	Cream apply	twice a day
Sodium Cromoglycate Eye Drops two drops four times a day	Sodium Cromoglycate Eye Drops	two	drops
Sudafed 12 Hours 120mg Tablet one tab twice a day	Sudafed 12 hours	120mg Tablet	one tab
Ventolin Nebules 2.5mg/2.5mL Solution one every four hours	Ventolin Nebules 2.5mg/2.5mL	Solution one	every four hours
Zolpidem Tablet ten mg before bed	Zolpidem Tablet ten	mg	before bed

During the second part of the evaluation, we examined the ability of our prescribing habit analysis process in identifying prescriptions that deviate from normal prescribing habits. The results from evaluating the first test set were organised into a contingency table as shown in Table 3. As described at the start of this section, 30 of the 190 prescrip-

Table 3. A contingency table summarising the results of evaluating the prescribing habit analysis process using the first test set.

		Gold Standard		
		Valid	Invalid	
Obtained Results	Valid	tp = 153	fp = 0	153
	Invalid	fn = 7	tn = 30	37
		160	30	190

tions in the first test set contain artificial medication errors. In other words, $fp + tn = 30$ prescriptions are considered as invalid. The remaining $tp + fn = 160$ prescriptions in the first set are valid prescriptions, or prescriptions that follow the conventional prescribing habits. From Table 3, we can observe that our prescribing habit analysis process achieved a 100% precision with an accuracy of 96.32% and a recall at 95.63% using the first test

set. In other words, all prescriptions determined as valid by our process are truly valid. On the other hand, about 4.38% of valid prescriptions were not detected (i.e. labelled as invalid) by our process. A closer look at the results showed that these 7 errors in validation is due to incorrectly bracketed prescriptions. In other words, about 7 in 10 or 70.00% of the problems with bracketing resulted in validation errors using the first test set.

Table 4. The additional 1 error that caused a drop in the bracketing performance using the second test set. Note that the drug name is incorrectly spelt. The correct spelling for “*Podophylotoxin*” is “*Podophyllotoxin*”.

Full Prescription String	Drug Name	Dosage	Frequency
Podophylotoxin 0.5% Paint 1 apply twice a day	Podophylotoxin 0.5%	Paint 1 apply	twice a day

For the third part of the evaluation, we performed the bracketing and analysis process on the second test set. Due to the spelling errors, the accuracy of the bracketing process dropped by 0.53% to 94.21% where 11 out of the 190 prescriptions were incorrectly structured. Out of the 11 errors, 10 were inherited from the first test set as described previously in Table 2. The additional 1 error in bracketing were caused by the spelling mistakes introduced into the second test set. Table 4 shows the 1 error. This 1 bracketing error due to spelling mistakes propagated to the prescribing habit analysis process. As summarised in Table 5, the additional 1 error caused a drop in the accuracy and the recall to 95.26% and 94.38%, respectively. Using this second test set, we observed that 30 prescriptions with spelling errors in a set of 190, have the potential of lowering the accuracy of prescribing habit analysis by 1.05%. Such performance deterioration is kept to the minimal by our prescription validation technique using the fuzzy string comparison process. Any approach based on exact string matching (e.g. the = operator), which produces either true or false matches, will suffer from greater degradation in the face of spelling errors.

Table 5. A contingency table summarising the results of evaluating the prescribing habit analysis process using the second test set. Note the rise in the number of fn from 7 in Table 3 to 9 due to the additional 1 bracketing error caused by the introduction of spelling mistakes into the second test set.

		Gold Standard		
		Valid	Invalid	
Obtained Results	Valid	tp = 151	fp = 0	151
	Invalid	fn = 9	tn = 30	39
		160	30	190

5. Limitations and future work

The proposed technique is able to validate free-text prescriptions with a relatively high level of accuracy above 95%. Considering the simplicity of the interface (i.e. minimal user involvement in setting system parameters), and the absence of the need for expertise to maintain complex knowledge bases, such performance is commendable. As part of our plan to deploy the validation technique through the MMEx system, we will evaluate the validation technique further using real-time data in clinical settings. The technique will be used to detect and prevent mishaps that arise from drug overdoses, incorrect frequencies, drug name confusions and spelling errors.

Another issue that requires more work is the technique’s lack of consideration for other types of less common information which can be present in prescription text. For example, during the prescribing process, the route for administering the drug (whether oral, intravenous, inhalation, subcutaneous, etc) and other information such as should the medication be take with food, etc may be present. While the current technique assumes the components of a prescription to be limited to three (i.e. drug name, dosage, frequency), we do acknowledge and have plans to extend the technique to cater for other information.

Another possible area for improvement is the introduction of a normalisation process to convert proprietary drug names into generic ones for improving the performance of the validation technique. In simple terms, the same drug may be referred to using different names, and at the moment, our validation technique do not take this into account. For instance, a drug as common as the analgesic paracetamol is referred to by many different proprietary names such as “*Tylenol*”, “*Panadol*”, “*Panamax*”, “*Perdolan*”, “*Calpol*”, “*Doliprane*”, etc. The use of fuzzy string comparison algorithms will not be able to counter such synonymy, and the presence of drug name variants can affect the performance of any techniques.

Lastly, we also have plans to extend the technique to perform retrospective validation of existing prescriptions in the databases of MMEx to identify past errors. The results of such retrospective validation will be an invaluable resource for auditing and quality improvement of health care institutions.

6. Conclusion

While only a small number of medication errors actually result in death, those that do usually have far reaching effects. Realising this, various laws, guidelines and technologies have been introduced in an attempt to eliminate medication errors, if not to lessen the effect of such mistakes. The technology of CPOE in particular, is already demonstrating varying level of success in inpatient settings. However, a host of concerns such as expertise, costs and infrastructures has continued to prevent these advanced technologies from seeing wider adoptions, especially amongst small and rural practices.

This research looked at ways for promoting safer prescribing amongst physicians. In this paper, we focused on three technical challenges that rendered complex CPOE systems infeasible for small and rural practices. They are (1) the problem of knowledge maintenance, (2) complex interface that disrupts work flow, and (3) intolerance to noisy prescription text. We proposed a prescription text validation technique that consists of a prescription bracketing process and a prescribing habit analysis process. The validation technique accepts free-text input via a simple interface, structures the prescription into components, compares the components against normal prescribing habits, and returns the results with any suggestions to the users.

We evaluated the validation technique using two test sets comprising 190 prescriptions each. Evaluation using the first test set that consists of 30 medication errors revealed an accuracy of 94.74% (for prescription bracketing) and 96.32% (for prescribing habit analysis). The bracketing process was able to correctly structure 180 prescriptions into their components. As for the analysis process, only 7 out of the 160 valid prescriptions were not detected, and all 30 invalid prescriptions were correctly identified. Evaluation

using the second set that consists of 30 spelling mistakes, in addition to the existing 30 medication errors, saw a slight drop in the accuracy to 94.21% (for bracketing) and 95.26% (for analysis).

All in all, we have introduced a robust technique for validating prescription text provided via a simple interface which is reminiscent of the conventional prescription pads. This technique is well suited for physicians of small and rural practices that require some form of decision support during the prescribing process without the overhead of CPOE systems. The technique can be used to detect and prevent mishaps that arise from drug overdoses, incorrect frequencies, drug name confusions and spelling errors. As we have shown, the technique does not require complex knowledge bases and interacts with the physicians through a simple user interface. Moreover, the accuracy of the technique degrades gracefully in the face of spelling errors.

References

- [1] K. Easton, T. Morgan, and M. Williamson. Medication safety in the community: A review of the literature. Report, National Prescribing Service, 2009.
- [2] C. Crowley, I. Maidment, M. Elswood, S. Conroy, Z. Davar, A. Fox, and G. Cavell. Safety in doses: Improving the use of medicines in the nhs. Report 0469/1007, National Patient Safety Agency, 2009.
- [3] L. Kohn, J. Corrigan, and M. Donaldson. To err is human: Building a safer health system. National Academy Press, Washington DC, 2000.
- [4] G. Miller, H. Britt, and L. Valenti. Adverse drug events in general practice patients in australia. *Medical Journal of Australia*, 184(7):321–324, 2006.
- [5] D. Bates, D. Cullen, N. Laird, L. Petersen, S. Small, D. Servi, G. Laffel, B. Sweitzer, B. Shea, and R. Hallisey. Incidence of adverse drug events and potential adverse drug events. *Journal of the American Medical Association*, 274(1):29–34, 1995.
- [6] R. Kaushal, D. Bates, C. Landrigan, K. McKenna, M. Clapp, F. Federico, and D. Goldmann. Medication errors and adverse drug events in pediatric inpatients. *Journal of the American Medical Association*, 285(16):2114–2120, 2001.
- [7] C. Zhan, R. Hicks, C. Blanchette, M. Keyes, and D. Cousins. Potential benefits and problems with computerized prescriber order entry. *American Journal of Health-System Pharmacy*, 63(4):353–358, 2006.
- [8] M. Lisby, L. Nielsen, and J. Mainz. Errors in the medication process: Frequency, type, and potential. *International Journal for Quality in Health Care*, 17(1):15–22, 2005.
- [9] E. Roughead and J. Lexchin. Adverse drug events: Counting is not enough, action is needed. *Medical Journal of Australia*, 184(7):315–316, 2006.
- [10] D. Bates. Using information technology to reduce rates of medication errors in hospitals. *British Medical Journal*, 320(7237):788–791, 2000.
- [11] E. Ammenwerth, P. Schnell-Inderst, C. Machan, and U. Siebert. The effect of electronic prescribing on medication errors and adverse drug events: A systematic review. *Journal of American Medical Informatics Association*, 15(5):585–600, 2008.
- [12] M. Reckmann, J. Westbrook, Y. Koh, C. Lo, and R. Day. Does computerized provider order entry reduce prescribing errors for hospital inpatients? a systematic review. *Journal of American Medical Informatics Association*, 16(5):613–623, 2009.
- [13] K. Shojania, B. Duncan, K. McDonald, and R. Wachter. Making health care safer: A critical analysis of patient safety practices. Report 01-E058, Agency for Healthcare Research and Quality, 2001.
- [14] T. Pryor, R. Gardner, P. Clayton, and H. Warner. The help system. *Journal of Medical Systems*, 7(2):87–102, 1983.
- [15] A. Geissbuhler and R. Miller. Wizorder, a user-friendly interface for order entry and clinical decision support tools. In *Proceedings of the 19th Annual Symposium on Computer Applications in Medical Care*, page 1002, Philadelphia, USA, 1995.

- [16] A. Geissbuhler and R. Miller. Distributing knowledge maintenance for clinical decision-support systems: The knowledge library model. In *Proceedings of the AMIA Annual Symposium*, pages 770–774, Washington DC, USA, 1999.
- [17] K. Clauson, W. Marsh, H. Polen, M. Seamon, and B. Ortiz. Clinical decision support tools: Analysis of online drug information databases. *BMC Medical Informatics and Decision Making*, 7(7):1–7, 2007.
- [18] A. Wright, D. Bates, B. Middleton, T. Hongsermeier, V. Kashyap, S. Thomas, and D. Sittig. Creating and sharing clinical decision support content with web 2.0: Issues and examples. *Journal of Biomedical Informatics*, 42(2):334–346, 2009.
- [19] L. Pizzi, D. Suh, J. Barone, and D. Nash. Factors related to physicians adoption of electronic prescribing: Results from a national survey. *American Journal of Medical Quality*, 20(1):22–32, 2005.
- [20] M. Kaye. Mandating of electronic prescriptions for medicare patients. *Online Journal of Nursing Informatics*, 12(Number 2):1–5, 2008.
- [21] K. Johnson and C. Weigle. Ambulatory computerized provider order entry. In C. Lehmann, G. Kim, and K. Johnson, editors, *Pediatric Informatics: Computer Applications in Child Health*. Springer, 2009.
- [22] P. Virk, D. Bates, J. Halamka, G. Fournier, and J. Rothschild. Analyzing transaction workflows in an eprescribing system. In *Proceedings of the AMIA Annual Symposium*, page 1129, Washington DC, USA, 2006.
- [23] D. Evans, N. Brownlow, W. Hersh, and E. Campbell. Automating concept identification in the electronic medical record: An experiment in extracting dosage information. In *Proceedings of the AMIA Annual Symposium*, pages 388–392, Washington DC, USA, 1996.
- [24] J. Shu, G. Clifford, W. Long, G. Moody, P. Szolovits, and R. Mark. An open-source, interactive java-based system for rapid encoding of significant events in the icu using the unified medical language system. In *Proceedings of the Annual Computers in Cardiology Conference*, pages 197–200, Chicago, USA, 2004.
- [25] A. Shah and C. Martinez. An algorithm to derive a numerical daily dose from unstructured text dosage instructions. *Pharmacoepidemiology and Drug Safety*, 15(3):161–166, 2006.
- [26] S. Gold, N. Elhadad, X. Zhu, J. Cimino, and G. Hripcsak. Extracting structured medication event information from discharge summaries. In *Proceedings of the AMIA Annual Symposium*, pages 237–241, Washington DC, USA, 2008.
- [27] J. Wilkinson, W. Arensman, N. Price, J. Madrid, and K. Pickens. Us patent no. 20090099870: Automated interpretation of medical prescription text, 2009.
- [28] R. Wagner and M. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.