Automating Concept Identification in the Electronic Medical Record: An Experiment in Extracting Dosage Information

David A. Evans, Ph.D.[†], Nicholas D. Brownlow[†], William R. Hersh, M.D.[‡], Emily M. Campbell, R.N.[‡]

[†]CLARITECH Corporation
319 South Craig St., Suite 200
Pittsburgh, Pennsylvania 15213

[‡]Oregon Health Sciences University
3181 S.W. Sam Jackson Park Road
Portland, OR 97201

We discuss the development and evaluation of an automated procedure for extracting drug-dosage information from clinical narratives. The process was developed rapidly using existing technology and resources, including categories of terms from UMLS96. Evaluations over a large training and smaller test set of medical records demonstrate an approximately 80% rate of exact and partial matches on target phrases, with few false positives and a modest rate of false negatives. The results suggest a strategy for automating general concept identification in electronic medical records.

INTRODUCTION

The value of electronic medical records is closely tied to our ability to find and transform the information they contain. In many medical records systems, the interpretation or coding of content is performed by humans and subsequent automated processing is often based on the manual annotations. Leaving aside the question of the independent value of such annotations, it is clear that the manual coding of records is a time-consuming (expensive) task that is prone to error—in accuracy, completeness, and consistency. A system that could automatically find and 'canonicalize' the relevant content of medical records would have an impact in reducing health-care costs and in improving the quality of clinical information.

Natural-language processing (NLP) remains one of the most promising, if least realized, approaches to the problem of managing the content of medical records. One subdiscipline of NLP, *information extraction (IE)*, is expressly focused on the task of identifying references to concepts in free text and transforming them into canonical codes, annotations, or database entries. The Message Understanding Conferences (MUC) have presented much general work in this area.[1] In the medical domain, a number of efforts have explored the parameters of the medical-language problem, contributed to our understanding of design requirements for effective NLP, and developed system prototypes, including those of the Linguistic String Project,[2] the MedSORT Project,[3] the Columbia Group's applications of the MED,[4] and others.[5,6] Increasingly, researchers are applying extraction techniques developed outside the medical domain to medical-record coding.[7,8,9]

Information-extraction techniques typically leverage word-level (or lexical-semantic) processing coupled with pattern (grammar) matching. They do not depend on global knowledge, deep domain semantics, or an ability to process discourse, though such resources, if available and reliable, could be exploited. Any IE module can be designed as a 'specialist'—one that can extract information of a very specific type, such as the signs and symptoms of particular diseases or the observations associated with chest radiology reports—and IE modules can be run independently of one another. Thus, it is possible to imagine a design for medical record processing consisting of some number of IE modules (possibly fewer than 100) working in parallel to extract all relevant information quickly and completely. The argument for such an approach would be made stronger if it could be shown that the cost in time, effort, and resources required to develop a single, specialist module was significantly less than that required to establish similar processing performance in a monolithic system.

Our general hypothesis is that effective, specialist IE modules can be created efficiently using readily available resources. To test this hypothesis for a specific case, we built an extraction module to identify drug-dosage phrases in clinical narrative text. As a computational framework, we used the NLP facilities of the CLARIT™ system and the pattern-matching grammar compilers of the CLARIT NameHunter extraction system. For our specific task, two additional resources had to be created *de novo*: the pattern rule set and a small lexicon of special forms. The module's drug lexicon was derived directly from the 1996 edition of the Unified Medical Language System (UMLS96).[10] By taking advantage of existing technology and resources, we were able to develop the module in a short period of time. We report on our methodology and the effectiveness of the module in the following sections.

0195—4210/96/$5.00 © 1996 AMIA, Inc.

METHODOLOGY

Our methodology requires five steps: (1) establishing a model or definition of the concept to be extracted (viz., drug dosage), (2) preparing data (medical text) for training and testing, (3) creating an IE module, including preparing new resources, (4) processing the data to extract dosage references, and (5) scoring results.

Defining the Concept

We take as given that there is an interesting well-defined clinical concept that can be briefly characterized as a drug dosage and that references to such a concept can be identified in clinical text. In fact, we find many examples in actual medical records that confirm our view and suggest the structure of a concept. The expressions are typically realized as discrete phrases with reasonably clear syntactic boundaries and an internal structure in which only certain types of information can occur. Figure 1 gives a sample of such expressions along with an analysis of the types of information that may be subsumed in the phrases.

Lortab	1–2 tabs	q. 4–6 hrs	p.r.n.	pain
drug	*dose-level*	*frequency*	*necessity*	*purpose*
Velosef	375 mg	p.o.	q.i.d.	x 4 days
drug	*dose-level*	*route*	*frequency*	*duration*
25 units of	NPH insulin			
dose-level	*drug*			

Figure 1: A Sample of Drug-Dosage Expressions

Generalizing over a large sample of dosage expressions, we can define the concept more precisely as an *object* and its allowable *attributes*. The object is a drug or *pharmacologic substance* as defined in UMLS96. Attributes can be any of the sub-expressions that play the roles given in Table 1. Within an expression, attributes can have multiple values or ranges of values (as in "Tylox b.i.d. to t.i.d.").

Refining the Definition. Note that the definition above—a *drug* plus one or more *attributes*—permits certain cases that are not necessarily drug dosages. For example, it is possible to refer to drugs in contexts other than therapeutic administration: "The patient refused p.o. Aspirin." Even though the phrase "p.o. Aspirin" includes a *drug* and a *route*, one cannot conclude from the sentence above that a drug was actually administered. (This example reflects the *mention* but not the *use* of a drug-dosage expression.)

Table 1: Drug-Dosage Expression Attributes

Attribute-Type	**Example**
Drug *Variant*	"double strength"
Dose-Level	"10 mg", "one tablet"
Frequency	"b.i.d."
Rate	"10 mg/kg", "10 mg/hr"
Duration	"times 10 days"
Necessity Modifier	"p.r.n."
Purpose Modifier	"for pain"
Quantity Dispensed	"10 dispensed"
Route	"by mouth"
Device	"by feeding tube"

To exclude most such cases, we constrained the definition further. To be considered a drug dosage, a concept must include not only a drug but also at least one *sufficient* attribute. From the list of attributes, we chose the following as sufficient: *dose-level, frequency, rate, duration, necessity, purpose*, and *quantity*.

Furthermore, to simplify text matching for our experiment, we excluded changes in drug dosage from the definition, such as "The Trazodone dose was increased from 75 mg to 100 mg." Such phrases occurred less frequently in our corpus than unchanging drug dosages.

Preparing Data

We prepared two non-overlapping sets of data: a training corpus for use by the extraction module developer and a test corpus for evaluation. By design, the module developer was blind to the test corpus.

Source Material. Both corpora were drawn from a collection of discharge summaries, dated from 1994 through early 1995, that came from the Oregon Health Sciences University (OHSU) hospital. To prepare the collection for research use, staff and patient names were obfuscated.

Training Corpus. 1,000 discharge summaries were randomly chosen from the OHSU collection. The module developer added mark-up tags to the resulting corpus, identifying all phrases referring to drug-dosage concepts meeting the constrained definition above.

Test Corpus. An additional 50 discharge summaries were randomly chosen from the OHSU collection. A medically trained consultant (a registered nurse) marked all drug-dosage phrases meeting the definition. (Table 2 gives a summary of corpus statistics.)

Creating an Extraction Module

Given the basic processing resources of the CLARIT system and the CLARIT NameHunter

Table 2: Corpus Statistics

	OHSU	Training	Test
Size	80.5 MB	5.5 MB	300 KB
Documents	14,841	1,000	50
Phrases (as defined)	n/a	5,190	212

modules, we focused most of our effort on the development of lexical resources and the target pattern-matching grammar.

CLARIT/NameHunter Processing. The CLARIT/NameHunter system combines NLP with regular-expression-based pattern matching. In particular, for the IE module we developed, the system includes the following processing stages:

1. *Tokenization:* The input is split into words and punctuation.
2. *Stemming:* Words are normalized to their root forms.
3. *Syntactic Category Assignment:* Word roots are looked up in lexicons to determine their parts of speech.
4. *Semantic Category Assigment:* Some word roots receive a semantic category which replaces their syntactic category.
5. *Pattern Matching:* Regular expressions, defined over syntactic and semantic categories, are applied to the categorized input. Phrases that match are marked with tags.

Lexicons. The CLARIT system lexicon contains more than 80,000 standard English word roots. For our experiment, we developed two additional special lexicons. The first contains various unusual words and abbreviations found in drug-dosage phrases (about 300 forms and their variants along with semantic types, including such items as "b.i.d.", "bolus", "capsule", etc.). The second was derived directly from UMLS96: it consists of all of the strings for concepts classified under the UMLS semantic category *pharmacologic substance*. These were automatically converted into a CLARIT-compatible format. There are nearly 68,000 UMLS strings in this category. After we eliminated variation in character case and hyphen placement, which are irrelevant for CLARIT processing, the resulting drug lexicon contained about 60,000 entries. The third lexicon consists of supplementary drug names—243 terms discovered in the training corpus that were not found in UMLS96 (including terms such as "advil", "baby aspirin", "cis-platin", "darvocet", "iron sulfate", "lithium", and "percodan").

Pattern Rules. We defined a set of about 50 rules in the CLARIT/NameHunter regular expression formalism. The rules are designed to match textual expressions of the object and attributes in the drug-dosage concept definition.

We created two versions of the rule set. The strict version required the textual expression of the drug object to be a UMLS *pharmacologic-substance* string. After noticing that a number of common drugs were absent from the UMLS list, we created a more lax rule set. In this set, the drug text could be either a UMLS string or any single word. The requirement that sufficient attributes be present limited the false positives matched by this rule set, and we were able to match all or part of a wider range of drug names.

A sample of the highest-level rules in the grammar is given in Figure 2. The patterns under "Found" give the essential phrase structure of the expression, referring to other constituents also defined by rules. For example, the Pre-Trigger rule is designed to match one of two possibilities: (a) a dose-level expression, such as "10 mg", followed optionally by the word "of", or (b) a necessity phrase such as "prn". The top-level Found rule can be minimally satisfied if the Pre-Trigger expression is followed by a drug name.

```
Top-Level
Found:
  Pre-Trigger Drug
   | Pre-Trigger? Drug C-Attr-Tight
   | Pre-Trigger? Drug C-Attr-Loose
Pre-Drug Material
Pre-Trigger:
  PreDosage of?
   | prn
Dosage
C-DRF-Trigger: # Sufficient to trigger
  C-Dosage C-RouteOrDevice?
   | C-NumOrDosage C-RouteOrDevice? C-Frequency
   | C-NumOrDosage C-Frequency C-RouteOrDevice?
C-DRF-Allow: # Not sufficient
  C-NumOrRange? C-RouteOrDevice?
C-Mod-Dur: # Modifier, Duration, or both
  C-Modifier C-Duration?
   | C-Duration C-Modifier?
C-Attr-Tight: # Triggers: Dosage, Frequency
  C-DRF-Trigger C-Mod-Dur?
C-Attr-Loose: # Triggers: Modifier, Duration
  C-DRF-Allow C-Mod-Dur
```

Figure 2: A Sample of Pattern Rules (Grammar)

Processing Medical Records

We ran the extraction module with the strict and lax rule sets over the training and test corpora. The module inserted its own distinct mark-up tags around the phrases it matched. We were then able to compare the machine mark-up against the reference mark-up placed by the developer and the medical consultant. A sample processed text, with mark-up in place, is given in Figure 3. A gold-standard (human-annotated) phrase is bracketed by "<rxg>" and "</rxg>", giving its start and end, respectively. The IE module's processing (mark-up) is bracketed by "<rx>" and "</rx>".

Discharge Medications:
1. <rxg><rx>Ciprofloxacin 500 milligrams po</rx> two times daily times seven days</rxg>.
2. <rxg><rx>Cyclosporin 130 milligrams po</rx> two times daily</rxg>.
3. <rxg><rx>Nifedipine-XL 30 milligrams po daily </rx></rxg>.
4. <rxg><rx>Vicodin one to two tablets po every four to six hours as needed for pain</rx> (dispense twenty, no refills)</rxg>.

Figure 3: A Sample of Processed/Marked-Up Text

Scoring the Results

We classified the results into four categories, as follows:

1. *Unanalyzeable:* Overlap relationship too complex to analyze, e.g.,
   <rxg><rx>2-gram potassium</rxg>, <rxg>2-gram</rx> <rx>sodium</rxg>, <rxg>80-gram</rx> protein</rxg>
2. *False Negatives:* Reference mark-up with no overlapping machine markup, e.g.,
   <rxg>Advil p.o. q.day (two p.o. b.i.d.)</rxg>
3. *False Positives:* Machine mark-up with no overlapping reference mark-up, e.g.,
   temperature 36.5, <rx>weight 63.6 kg</rx>
4. *Matches:* Machine mark-up bracketed by reference mark-up, e.g.,
   <rxg><rx>Lortab 1–2 tablets q. 4-6 hours p.r.n. pain</rx></rxg>

RESULTS

Tables 3 and 4 give the counts and percentages of the classified results over the training and test corpora, respectively. In each case, we identify three sets of results: (1) SG: the strict rule set with UMLS drug lexicon, (2) LG: the lax rule set with UMLS drug lexicon, and (3) LGS: the lax rule set with UMLS and supplemental drug lexicon.

Table 3: Processing Results for Training Corpus

Strict Grammar (SG)		
Total Sequences:	5,160	(100%)
Unanalyzeable Sequences:	42	(0.8%)
False Negatives:	1,520	(29.5%)
False Positives:	13	(0.3%)
Total Hits (Exact and Partial):	3,585	(69.5%)
Total Exact Hits:	*2,720*	*(52.7%)*
Total Partial Hits:	*865*	*(16.8%)*
Lax Grammar (LG)		
Total Sequences:	5,262	(100%)
Unanalyzeable Sequences:	62	(1.2%)
False Negatives:	891	(16.9%)
False Positives:	128	(2.4%)
Total Hits (Exact and Partial):	4,181	(79.5%)
Total Exact Hits:	*3,016*	*(57.3%)*
Total Partial Hits:	*1,165*	*(22.1%)*
Lax Grammar/Supplemental Lexicon (LGS)		
Total Sequences:	5,282	(100%)
Unanalyzeable Sequences:	63	(1.2%)
False Negatives:	831	(15.7%)
False Positives:	149	(2.8%)
Total Hits (Exact and Partial):	4,239	(80.3%)
Total Exact Hits:	*3,137*	*(59.4%)*
Total Partial Hits:	*1,102*	*(20.9%)*

Table 4: Processing Results for Test Corpus

Strict Grammar (SG)		
Total Sequences:	214	(100%)
Unanalyzeable Sequences:	4	(1.9%)
False Negatives:	68	(31.8%)
False Positives:	6	(2.8%)
Total Hits (Exact and Partial):	136	(63.6%)
Total Exact Hits:	*106*	*(49.5%)*
Total Partial Hits:	*30*	*(14.0%)*
Lax Grammar (LG)		
Total Sequences:	218	(100%)
Unanalyzeable Sequences:	4	(1.8%)
False Negatives:	32	(14.7%)
False Positives:	10	(4.6%)
Total Hits (Exact and Partial):	172	(78.9%)
Total Exact Hits:	*132*	*(60.6%)*
Total Partial Hits:	*40*	*(18.3%)*
Lax Grammar/Supplemental Lexicon (LGS)		
Total Sequences:	222	(100%)
Unanalyzeable Sequences:	4	(1.8%)
False Negatives:	28	(12.6%)
False Positives:	14	(6.3%)
Total Hits (Exact and Partial):	176	(79.3%)
Total Exact Hits:	*137*	*(61.7%)*
Total Partial Hits:	*39*	*(17.6%)*

In general, we see parallel results for both the training and test corpora. The number of unanalyzeable sequences is small; the number of false positives is near trivial. The hit rate is very high (when compared to similar IE tasks in other domains) and increases with the lax grammar without a significant increase in false positives.

With the SG, false positives are almost all due to loose parsing of symptom attributes. For example, "Ethanol withdrawal seizures" is interpreted as "Ethanol [for] withdrawal seizures".

The LG yields a 10% increase in total hits—phrases containing drug names not in UMLS96. The LG accepts a single noun or unknown word as a *maybe-drug* if accompanied by sufficient attributes. For single-word drug names, this can yield a total match: "<rxg><rx>Depakote 500 mg p.o. t.i.d.</rx></rxg>". If the name is complex, we get the rightmost word: "<rxg>Potter's <rx>cocktail 400 cc</rx> to take 30–40 cc q.4–6h. p.r.n. pain</rxt>".

Most of the LG's false positives (93 of 128, or 73%) are people's weights: "male, 3500 grams"; "weight 10.05 kg". Here the words "male" and "weight" are interpreted as *maybe-drugs*. Such false positives could be blocked by requiring additional attributes besides the dosage level.

The LGS yields a small (0.8%) increase in total hits. With the supplementary drug lexicon, the extraction module has almost all of the drug names found in the corpus. This yields a few more hits than the LG because the attribute parsing rules for phrases containing known drugs are looser than those for *maybe-drugs*.

CONCLUSION

Our work demonstrates that it is possible to produce a reasonably accurate, semantically focused extraction module in a short period of time. Corpus preparation and mark-up took about 28-person-hours; development of general tools (for corpus management and diagnostics), 31; and grammar and resource development, 45.

This result suggests a strategy for automating concept identification in medical records. Rather than attempt to develop a comprehensive system for semantic understanding of clinical text (if such were even possible), we might more efficiently develop individual modules for different kinds of concepts. Further explorations of this strategy might include experiments in the development of other IE modules and with the integration and interaction of different modules.

Acknowledgements

Our work was supported, in part, by award #LM-05879 from the National Library of Medicine.

References

1. *Proceedings of the Fifth Message Understanding Conference (MUC-5).* Baltimore (MD): Morgan Kaufman, 1993.
2. Sager N, Friedman C, Lyman MS. *Medical Language Processing: Computer Management of Narrative Data.* Menlo Park (CA): Addison-Wesley, 1987.
3. Evans DA: *Final Report on the MedSORT-II Project: Developing and Managing Medical Thesauri.* Technical Report No. CMU-LCL-87-3, Laboratory for Computational Linguistics, Carnegie Mellon University. 1987.
4. Friedman C, Alderson PO, Austin JHM, Cimino JJ, Johnson SB. A general natural-language text processor for clinical radiology. *J Am Med Inform Assoc* 1994; 1(2):161–174.
5. Canfield K, Bray B, Huff SM, Warner HR. Database capture of natural language echocardiology reports. In: *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care.* Washington (DC): IEEE Society Press, 1989:559–563.
6. Haug PJ, Ranum DL, Frederick PR. Computerized extraction of coded findings from free-text radiology reports. *Radiology* 1990; 174:543–548.
7. Aronow DB, Cooley JR, Soderland S. Automated identification of episodes of asthma exacerbation for quality measurement in a computer-based medical record. In: *Proceedings of the Nineteenth Annual Symposium on Computer Applications in Medical Care.* Philadelphia (PA): Hanley & Belfus, 1995:309–313.
8. Lehnert W, Soderland S, Aronow D, Feng F, Shmueli A. Inductive text classification for medical applications. *Journal for Experimental and Theoretical Artificial Intelligence* 1995; 7:49–80.
9. Soderland S, Aronow D, Fisher D, Aseltine J, Lehnert W. Machine learning of text analysis rules for clinical records. Amherst (MA): CIIR;1995 Technical Report: TE-39.
10. *UMLS Knowledge Sources.* National Library of Medicine, 7th experimental edition. 1996.