MEDINFO 2010
C. Safran et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-588-4-944

Automatically Detecting Medications and the Reason for their Prescription in Clinical Narrative Text Documents

Stéphane M. Meystre, Julien Thibault, Shuying Shen, John F. Hurdle, Brett R. South

Department of Biomedical Informatics, University of Utah, Salt Lake City, Utah, USA

Abstract

An important proportion of the information about the medications a patient is taking is mentioned only in narrative text in the electronic health record. Automated information extraction can make this information accessible for decision-support, research, or any other automated processing. In the context of the "i2b2 medication extraction challenge," we have developed a new NLP application called Textractor to automatically extract medications and details about them (e.g., dosage, frequency, reason for their prescription). This application and its evaluation with part of the reference standard for this "challenge" are presented here, along with an analysis of the development of this reference standard. During this evaluation, Textractor reached a system-level overall F_1-measure, the reference metric for this challenge, of about 77% for exact matches. The best performance was measured with medication routes (F_1-measure 86.4%), and the worst with prescription reasons (F_1-measure 29%). These results are consistent with the agreement observed between human annotators when developing the reference standard, and with other published research.

Keywords:

Pharmaceutical preparations, Drug prescriptions, Natural language processing, Program evaluation, Knowledge bases.

Introduction

Computerized physician order-entry (CPOE) and E-Prescribing systems are becoming widely available in the healthcare system [1], and provide detailed information about the medications prescribed and managed with these systems, but a substantial proportion of the medications actually taken by the patient are still only mentioned in narrative clinical text documents in the patient electronic health record. These medications were prescribed in another institution or private practice, were bought over-the-counter, or were prescribed before the introduction of CPOE. Their mention in narrative text format makes them inaccessible for decision-support, research, or any other automated processing. These functionalities require coded data, and as a possible answer to this issue, Natural Language Processing (NLP) can convert narrative text into coded data. Techniques for automatically encoding textual documents from the electronic health record have been evaluated by several groups, as described in Meystre et al. [2]. Examples are the Linguistic String Project [3] and MedLEE (Medical Language Extraction and Encoding system) [4]. Other systems automatically mapping clinical text concepts to standardized vocabularies have been reported, such as MetaMap [5]. MetaMap and its Java™ version called MMTx (MetaMap Transfer) were developed by the U.S. National Library of Medicine. MetaMap has been shown to identify most concepts present in MEDLINE titles [6] and has been used for Information Retrieval [7] and Information Extraction [8].

When extracting information from narrative clinical text documents, the context of the extracted concepts plays a critical role. Important contextual information includes negation (e.g., "denies any chest pain"), temporality (e.g., "...fracture of the tibia 2 years ago..."), and the event subject identification (e.g., "his mother has diabetes"). NLP systems such as the LSP [3] or MedLEE [4] include negation analysis in their processing, but research focused explicitly on negation detection started only a few years ago with algorithms like NegEx [9].

The automated extraction of information from clinical text documents has been the focus of several competitions — called "challenges" — these last few years. Prominent ones were organized by the i2b2 (Informatics for Integrating Biology and the Bedside) National Center for Biomedical Computing. These "challenges" took the recent application of NLP to clinical research a step further by providing a de-identified corpus of clinical narrative text documents and by stimulating new developments in this domain. The i2b2 "challenges" started in 2006, with an automated de-identification challenge [10], and a smoking status detection challenge [11]. The obesity challenge was organized in 2008 [12]. The latest "i2b2 challenge" was organized in 2009 and focused on the extraction of medications, details about these medications, and reasons for their prescription. A corpus of 1249 clinical text documents (discharge summaries from Partners Healthcare in Boston, MA) has been semi-automatically de-identified and re-identified with realistic surrogates, and then split into a training corpus of 696 documents, and a test corpus of 553 documents. Only 17 documents in the training corpus were annotated by the organizing team for medications and prescription details; all other documents in the training corpus were not annotated.

For this challenge, we built a new NLP application based on the UIMA (Unstructured Information Management Architecture) framework [13]. This new application, called Textractor, and its evaluation using the first part of the reference standard are presented here.

Materials and Methods

Architecture and development process

Text analysis functionalities developed for Textractor are implemented as modules organized in a pipeline, as depicted in Fig. 1. The whole analysis pipeline is implemented in the UIMA framework. UIMA provides a development model that enforces the use of XML description files for maintainability and interoperability, as well as tools to test and visualize the text annotations realized by the system.

The pre-processing phase starts with the analysis of the document structure. Each document is broken into sections using regular expressions to match section titles or subtitles. Some sections that typically contain mentions of medications that should not be extracted by our system are filtered out (e.g., "Family history" sections mention drugs taken by family members, "Allergies" sections mention drug allergies that should not be extracted for this i2b2 challenge). For each section of interest, the text is split in sentences using an OpenNLP [14] module based on the maximum entropy principle. The whole text is then tokenized, and a part-of-speech (POS) tagger is applied to the set of tokens. The POS tagger is also based on an OpenNLP module, with a supplementary logic to treat sections where the end of the sentence cannot be inferred without knowledge of the section content structure.

During the second phase, medications and details about them (dosage, duration, frequency, route, reason for prescription) are extracted. We use MMTx to extract medications and possible reasons for their prescription. MMTx was developed to extract data from MEDLINE abstracts, and acronyms are less common in paper abstracts than in clinical documents, and are the principal source of ambiguity for our system. Examples of acronyms ambiguous to MMTx are "Dr." (detected as diabetic retinopathy), "Mr." (mitral regurgitation), "M.D." (mental depression), etc. For disambiguation, we expand abbreviations and acronyms before feeding MMTx with each sentence of the document to parse. A list of abbreviations and acronyms and corresponding full terms from the APL system [15] was expanded for this purpose. MMTx (version 2.4.C) is used to extract UMLS Metathesaurus [16] concepts related to drugs and health conditions. More specifically, the system implements the MMTxAPILite class and uses the default dataset (complete 2006 UMLS Metathesaurus) and settings. The following semantic types were used to extract medications: Amino Acid, Peptide, or Protein (aapp), Antibiotic (antb), Biologically Active Substance (bacs), Carbohydrate (carb), Hormone (horm), Organic Chemical (orch), Pharmacologic Substance (phsu), Steroid (strd), Vitamin (vita), and the following semantic types for possible prescription reasons: Disease or Syndrome (dsyn), Congenital Abnormality (cgab), Finding (fndg), Pathologic Function (patf), Sign or Symptom (sosy), Therapeutic or Preventive Procedure (topp). Since MMTx lacks context analysis (e.g., Insulin will be extracted in "…glucose management didn't require insulin …"), a context analysis step is also required after the extraction. Context analysis is based on an improved version of NegEx. This algorithm uses regular expressions and lists of terms to analyze negation (a concept can be affirmed, negated, or possible). Our implementation uses a variable window to analyze the context of each concept (instead of the 5 tokens fixed window originally used in NegEx) and infers the context from a larger set of base phrases. Finally, medication attributes (dose, frequency, duration, and route) are extracted with a set of regular expressions.

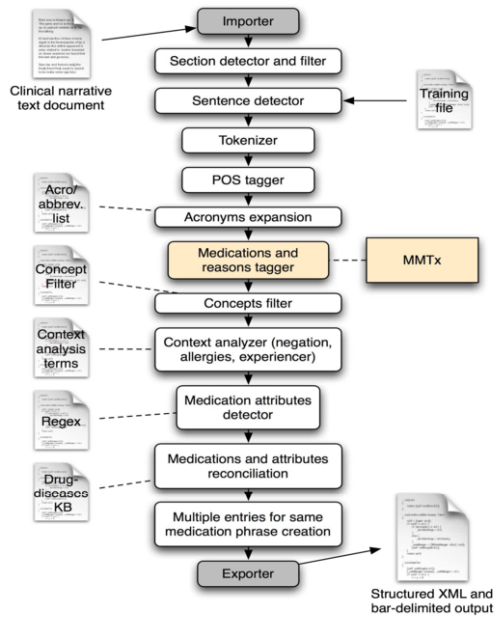

Figure 1- Components of Textractor for medications extraction

During the last phase, extracted medications are combined with medication attributes to build the prescription annotations. Each medication is combined with attributes that follow (or sometimes precede) it according to a set of rules. For example, in "Rheumatology suggested starting colchicine 0.6 mg b.i.d. for two days"…, the medication *colchicine* is followed by a *0.6 mg* dose, a *b.i.d.* frequency, and a *for two days* duration that are combined with the medication. Reason for prescription annotations are more complicated. As mentioned earlier, possible reasons are extracted with MMTx. They are then linked with the corresponding medication using a few rules and regular expressions that recognize possible reasons preceded by expressions like "because of …", "due to …", "… was treated with", etc. For the prescription annotations that do not have any reason attribute after applying this logic, we complement the search with the use of a drug-disease knowledge base. This knowledge base was built from existing databases that include the Pharmacogenomics Knowledge Base

(PharmGKB; available at www.pharmgkb.org), the Comparative Toxicogenomics Database (CTD; available at ctd.mdibl.org), and the UMLS Metathesaurus. The final knowledge base contains about 750 paired relationships of medication and disease, with their CUIs, and their relationship type. If a possible reason is found in a window of ± 2 lines of a medication (as defined for the i2b2 challenge) and is related to this medication according to the knowledge base, then it is added as the reason attribute for this prescription.

Finally, multiple prescription annotations are created for a single medication if the associated attributes describe multiple values or reasons (e.g., "Tylenol 650 mg p.r.n. pain or headache" becomes two *Tylenol – 650 mg – p.r.n.* annotations, one with the reason *pain*, and the other with the reason *headache*).

Reference standard creation

As mentioned previously, part of the reference standard (251 documents) was built by all teams participating in the challenge. Each document was annotated by one member of two different teams participating in the challenge with a member of a third team adjudicating disagreements in a second step. This process produced a final reference standard created by the challenge teams that could be used for evaluation purposes. Assigning annotation tasks to challenge participants is one of the novel approaches for this i2b2 challenge.

For our own annotations, we created an annotation schema using an open source annotation tool called Knowtator [17] and based on the annotation guideline provided by the i2b2 challenge team. Knowtator is a Protégé [18] plugin tool that uses the unique knowledge representation capabilities of Protégé to develop complex annotation schemas. Our annotation schema treats medication as the parent class and all other related information as child subclasses. Each medication class has an associated slot attribute describing whether the annotated mention was found in a list or in narrative text, and a complex slot attribute used to link annotated subclass information with the parent medication class. Using the Knowtator tool and this annotation schema, the span of medication mentions can easily be annotated and linked with associated spanned mentions of dose, route, frequency, duration, and reason.

Our team was assigned 40 reference standard documents for annotation. From these documents, each member of our team was assigned 10 documents to annotate using the guideline and the Knowtator annotation schema (10 documents were annotated by two of us, for subsequent agreement analysis, as described below). The logic for annotation tasks for each mention of medication in the clinical texts was as follows: a) identify the parent class medication; b) determine if the identified mention is in the context of a list or narrative test; c) identify associated subclass mentions of dose, route frequency, duration, and/or reason; d) link the subclass mentions with the parent class medication; e) identify the next medication mention (Figure 2).

Due to the complexity of this challenge, we felt it was necessary to evaluate the performance of human beings on annotation tasks related to this challenge. We evaluated reliability (task consistency) of the team annotation task using a subset of 10 documents from the assigned document set. Two team members annotated each of these documents. Logical pairings were created so that each annotator was evaluated against every other annotator on our team. Task consistency was evaluated using inter-annotator agreement, as published by Roberts [19], using the formula for the Inter Annotator Agreement (IAA): IAA = matches / (matches + non-matches).

We report IAA for class, subclass and slot attribute agreement for instances where class matched with an overlapping span, or where class and span matched exactly.

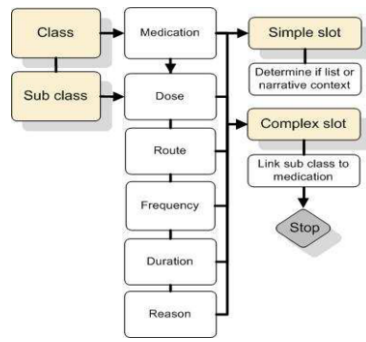

Figure 2- Annotation task logic process flow diagram

Medication extraction evaluation

Evaluation for this "i2b2 medication extraction challenge" is realized with exact matches (i.e. the extracted phrase corresponds exactly to the reference standard) and with inexact matches (i.e. the extracted phrase overlaps with the reference standard) separately.

Exact matches evaluation is done at the instance level, and metrics are recall (number of correct distinct instances extracted / all instances in the reference standard), precision (number of correct distinct instances extracted / all instances extracted), and the F1-measure (a harmonic mean of recall and precision, with a weight of 1 [20]). Instances that are not mentioned are ignored.

Inexact matches evaluation is realized at the token level (i.e. words or character groups separated by a white space). Metrics are also recall, precision, and F1-measure, but recall and precision are calculated differently (recall = number of correct tokens extracted / all tokens on the reference standard; precision = number of correct tokens extracted / all tokens extracted).

Exact and inexact matches evaluations are done separately for each class (e.g., all doses or all medications) and also for each full prescription annotation (i.e. medication, prescription details, and reason for the prescription if present), and then averaged at the document level or at the system level (i.e. over all medications extracted). For the example in Table 1, these metrics would be: exact recall = 3/5 (3 correct instances extracted and 5 instances in the reference standard); exact precision = 3/5 (3 correct instances extracted and 5 instances extracted); inexact recall = 5/8 (5 correct tokens extracted: toprol, 50, mg,

p.o., b.i.d.; 8 tokens in the reference standard); inexact precision = 5/6 (5 correct tokens extracted: toprol, 50, mg, p.o., b.i.d.; 6 tokens extracted).

Table 1- Medication extraction example

Medication instance extracted					
Medication	Dose	Route	Frequency	Duration	Reason
toprol	50 mg	p.o.	b.i.d.	Nm	Asthma
Reference standard					
Medication	Dose	Route	Frequency	Duration	Reason
toprol xl	50 mg	p.o.	b.i.d.	3 weeks	Nm

nm = not mentioned

Results

The testing corpus of 553 documents was made available for three days in August 2009, and each participating team could submit up to three runs. We analyzed this corpus with three slightly different configurations of Textractor: the first included the 15 UMLS semantic types cited above, the second had fewer prescription reason semantic types (aapp, antb, carb, horm, orch, phsu, strd, bacs, vita and dsyn, patf, sosy), and the third fewer medication and prescription reason types (antb, phsu, vita and dsyn, patf, sosy).

Local evaluation details and results

The results of our first configuration of Textractor, with the part of the reference standard annotated by the participating teams, are presented here in Tables 2 and 3. System-level results were averaged over all medications extracted by the system; patient-level results were averaged at the level of documents (since we had one document per patient).

Table 2-Results of the exact match evaluation

Information	**N**	**Syst R**	**Syst P**	**Syst F**	**Pat F**
Medication	8882	0.752	0.769	0.761	0.759
Dose	4432	0.758	0.910	0.827	0.811
Route	3417	0.813	0.921	0.864	0.842
Frequency	4074	0.781	0.890	0.832	0.824
Duration	545	0.329	0.395	0.359	0.347
Reason	1529	0.185	0.679	0.290	0.259
OVERALL	22879	0.720	0.830	0.771	0.760

Syst = system-level results; Pat = patient-level results;
R = recall; P = precision; F = F1-measure
N = number of instances of each class in the reference standard

Table 3- Results of the inexact match evaluation

Information	**N**	**Syst R**	**Syst P**	**Syst F**	**Pat F**
Medication	8882	0.766	0.782	0.774	0.784
Dose	4432	0.785	0.921	0.848	0.830
Route	3417	0.798	0.927	0.858	0.837
Frequency	4074	0.736	0.924	0.820	0.817
Duration	545	0.326	0.481	0.388	0.398
Reason	1529	0.145	0.697	0.240	0.244
OVERALL	22879	0.693	0.837	0.758	0.750

This corpus included 251 documents and took Textractor an average of about 24 seconds to analyze each document. Most of the time was spent extracting concepts with MMTx, and even when limiting the semantic types and skipping sections of the document for MMTx analysis, the concept extraction phase represented most of the execution time.

Annotation task reliability evaluation

For the 10 documents we used to evaluate task consistency, overall inter-annotator agreement for class and span exact matches (or with matching using overlapping span) was the highest for identification of mentions of medication 85.9% (92.4% partial match), and the lowest for identification of duration 16% (29.3%) (Table 4). Slot attribute agreement for overall exact match to determine if the medication was mentioned in a list or in narrative text was 62%, and 63% for linking subclass attributes with the parent medication class.

Table 4- Inter-annotator agreement (all values are percentages)

Annotation class/subclass	**N**	**Exact match IAA (class match, Span match)**	**Partial match IAA (class match, span overlap)**
Medication	303	85.9	92.4
Dosage	109	88.4	88.4
Route	119	76.5	81.0
Frequency	88	73.3	89.9
Duration	16	16.0	29.3
Reason	59	31.3	73.1
OVERALL	694	78.5	86.6

N = number of annotated instances of each class

Discussion

This evaluation showed that the NLP application we developed for this task – Textractor – performed satisfactorily. The reference metric for this challenge, the system-level overall F_1-measure, reached about 77% for exact matches. Performance was good for medication attributes like dose, route, and frequency, with recalls around 80% and precisions around 90%. Results were not as good for durations, with recall and precision between 30% and 40%, and for reasons, with a recall below 20%, and a precision below 70%. These two attributes are very difficult to define precisely and also resulted in low agreement when analyzing our own manual annotations. The exact match IAA is equivalent to the F_1-measure when scoring one annotator against the other (i.e., treating one annotation as reference and the other as test), and in our case, this agreement only reached 16% with durations and 31.3% with reasons for prescription, in similar ranges than the measured performance of Textractor.

Our results are also consistent with other published similar research, such as the MERKI system [21] with measured precisions of 83.7% for dose, 88% for route, and 83.2% for frequency. What distinguishes our work from MERKI is its foundation on an open-source pipeline, a more comprehensive test set, and broader multi-reviewer evaluation scheme.

The adoption of UIMA as a firm ground for our developments gave us several advantages: efficient development tools to test and visualize the results of the system, good integration with

Eclipse [22], use of standard XML description files for maintainability and interoperability, and easier integration of existing developments (e.g., OpenNLP tools). We also integrated MMTx in UIMA, and benefited from its good UMLS Metathesaurus concepts indexing. However, the significant pre- and post-processing required to use this application with clinical text, its relatively slow performance, the impossibility to adapt it for multi-processing, its planned phase-out by the NLM, and its lack of an API will lead us into the development of a new concept extraction tool integrated in UIMA.

For our annotations, considering the complexity of this annotation task, we were not surprised to see that exact span matching had much lower agreement compared with overlapping span matching for all classes of annotated information. Prevalence of annotated classes varied widely across the 10 documents used to assess annotator agreement at the class and subclass level. The high variability in the observed agreement could be partly explained by this small sample. A more formal evaluation of both reliability and validity of annotation tasks across the challenge in general would show interesting results and would help define how to deal with some of these issues.

The automated extraction of information from biomedical text is still a relatively new field of research, and the extraction of information from clinical text is even newer [2]. The potential uses of information extracted from clinical text are numerous and far-reaching. In the same way the Message Understanding Conferences have fostered the development of information extraction in the general domain, similar competitive challenges for information extraction from clinical text, such as the "i2b2 medications extraction challenge," will undoubtedly stimulate advances in the biomedical field.

Acknowledgments

We would like to thank the i2b2 challenge team for the development of the training and testing corpora and for the excellent organization of this challenge.

References

[1] Enterprises K. CPOE Digest 2006. Orem, Utah: KLAS Enterprises2009.

[2] Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. Extracting information from textual documents in the electronic health record: a review of recent research. Yearb Med Inform. 2008:128-44.

[3] Chi E, Lyman M, Sager N, Friedman C, editors. Database of computer-structured narrative: methods of computing complex relations. SCAMC 85; 1985.

[4] Friedman C, Johnson SB, Forman B, Starren J. Architectural requirements for a multipurpose natural language processor in the clinical environment. Proc Annu Symp Comput Appl Med Care. 1995:347-51.

[5] Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp. 2001:17-21.

[6] Pratt W, Yetisgen-Yildiz M. A Study of Biomedical Concept Identification: MetaMap vs. People. Proc AMIA Symp. 2003:529-33.

[7] Aronson AR. Query expansion using the UMLS Metathesaurus. Proc AMIA Symp; 1997.

[8] Weeber M, Klein H, Aronson AR, Mork JG, de Jong-van den Berg LT, Vos R. Text-based discovery in biomedicine: the architecture of the DAD-system. Proc AMIA Symp. 2000:903-7.

[9] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform 2001. p. 301-10.

[10] Uzuner O, Luo Y, Szolovits P. Evaluating the state-of-the-art in automatic de-identification. JAMIA. 2007 Sep-Oct;14(5):550-63.

[11] Uzuner O, Goldstein I, Luo Y, Kohane I. Identifying patient smoking status from medical discharge records. JAMIA. 2008 Jan-Feb;15(1):14-24.

[12] Uzuner O. Recognizing obesity and comorbidities in sparse data. JAMIA. 2009 Jul-Aug;16(4):561-70.

[13] Apache. UIMA (Unstructured Information Management Architecture). 2008; Available from: http://incubator.apache.org/uima/

[14] OpenNLP. 2009; http://opennlp.sourceforge.net/.

[15] Meystre S, Haug PJ. Automation of a problem list using natural language processing. BMC medical informatics and decision making 2005. p. 30.

[16] Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. Methods Inf Med. 1993;32:281-91.

[17] Ogren PV, Savova G, Buntrock JD, Chute CG. Building and evaluating annotated corpora for medical NLP systems. AMIA Annual Symp proceedings 2006 p. 1050

[18] Musen MA, Eriksson H, Gennari JH, Tu SW, Puerta AR. PROTEGE-II: a suite of tools for development of intelligent systems from reusable components. Proc Annu Symp Comput Appl Med Care. 1994:1065.

[19] Roberts A, Gaizauskas R, Hepple M, Davis N, Demetriou G, Guo Y, et al. The CLEF corpus: semantic annotation of clinical text. AMIA Annual Symp proc 2007:625-9.

[20] van Rijsbergen CJ. Information Retrieval: Butterworth; 1979.

[21] Gold S, Elhadad N, Zhu X, Cimino JJ, Hripcsak G. Extracting structured medication event information from discharge summaries. AMIA Annual Symp proc 2008:237-41.

[22] Eclipse IDE. 2009; http://www.eclipse.org/.

Address for correspondence:

Stephane M. Meystre, MD, PhD.
University of Utah, Department of Biomedical Informatics
26 S 2000 E, HSEB suite 5700 Salt Lake City, UT 84112, USA
E-mail: stephane.meystre@hsc.utah.edu Tel.: +1-801-581-4080