Textractor: a hybrid system for medications and reason for their prescription extraction from clinical text documents

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

► Additional tables and appendix are published online only. To view these files please visit the journal online (http://jamia.bmj.com).

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

Correspondence to
Dr Stephane M Meystre, University of Utah, Department of Biomedical Informatics, 26 S 2000 E, HSEB suite 5700, Salt Lake City, UT 84112, USA; stephane.meystre@hsc.utah.edu

Received 22 February 2010
Accepted 25 June 2010

ABSTRACT

Objective To describe a new medication information extraction system—Textractor—developed for the 'i2b2 medication extraction challenge'. The development, functionalities, and official evaluation of the system are detailed.

Design Textractor is based on the Apache Unstructured Information Management Architecture (UMIA) framework, and uses methods that are a hybrid between machine learning and pattern matching. Two modules in the system are based on machine learning algorithms, while other modules use regular expressions, rules, and dictionaries, and one module embeds MetaMap Transfer.

Measurements The official evaluation was based on a reference standard of 251 discharge summaries annotated by all teams participating in the challenge. The metrics used were recall, precision, and the F_1-measure. They were calculated with exact and inexact matches, and were averaged at the level of systems and documents.

Results The reference metric for this challenge, the system-level overall F_1-measure, reached about 77% for exact matches, with a recall of 72% and a precision of 83%. Performance was the best with route information (F_1-measure about 86%), and was good for dosage and frequency information, with F_1-measures of about 82—85%. Results were not as good for durations, with F_1-measures of 36—39%, and for reasons, with F_1-measures of 24—27%.

Conclusion The official evaluation of Textractor for the i2b2 medication extraction challenge demonstrated satisfactory performance. This system was among the 10 best performing systems in this challenge.

INTRODUCTION

Medical errors have been identified as the cause of numerous deaths, and even if some are difficult to avoid, many could be prevented. These preventable errors have been estimated to cause 100 000 deaths every year in the USA.[1] Among these preventable deaths in the USA, about 7000 can be associated with medication errors.[2] Computerized provider order-entry systems have been proposed to reduce this risk of medication errors. Such systems reduce errors when they provide decision support, and are becoming widely available in the healthcare system,[3] but a substantial proportion of the clinical information that their decision-support features could rely on (eg, medications the patient is taking, allergies, diseases) are still only mentioned in narrative clinical text documents in the patient electronic health record (EHR). Their mention in narrative text format makes them inaccessible for decision-support, research, or any other automated processing since these functionalities require coded data. A possible solution to this issue, natural language processing (NLP), can be used to automatically extract structured and coded information from narrative text.

To stimulate research and development in this domain, the Informatics for Integrating Biology and the Bedside (i2b2) National Center for Biomedical Computing (Boston, Massachusetts, USA) organized the 'i2b2 medication extraction challenge.' We participated in the latter and built a new information extraction system based on the Apache Unstructured Information Management Architecture (UIMA) framework.[4] This new application, called Textractor, its development, its evaluation during the 'challenge,' and the analysis of the errors it made are presented here, and are also available in more detail in the online-only appendix available at http://jamia.bmj.com.

Methods and resources to automatically extract clinical information from documents in the EHR have been evaluated by several groups, as described by Meystre *et al*[5] and Uzuner *et al*.[6] An example resource we used in the Textractor system is MetaMap Transfer (MMTx), the Java version of MetaMap, developed by the US National Library of Medicine, and used to map concepts in the analyzed text to Unified Medical Language System (UMLS) Metathesaurus[7] concepts.

MATERIALS AND METHODS

Information extraction task and clinical text corpus

The general objective of the i2b2 medication extraction challenge was to extract the list of medications found in patient clinical documents. This challenge, the task, the clinical text corpus and its annotation, and the evaluation metrics, are all described in detail in Uzuner *et al*.[6]

For our team annotations, we developed an annotation schema based on the guideline provided for this challenge using an open source annotation tool called Knowtator,[8] a plug-in for the Protégé[9] knowledge management system. In our annotation schema depicted in figure 1, the medication name is treated as the parent 'class' with other related information treated as children 'subclasses'. Two attributes referred to as 'slots' are associated with each medication name: a slot describing whether the annotated text was found in a list or in narrative text, and a second complex slot used to link annotated subclass information with the parent medication name class.

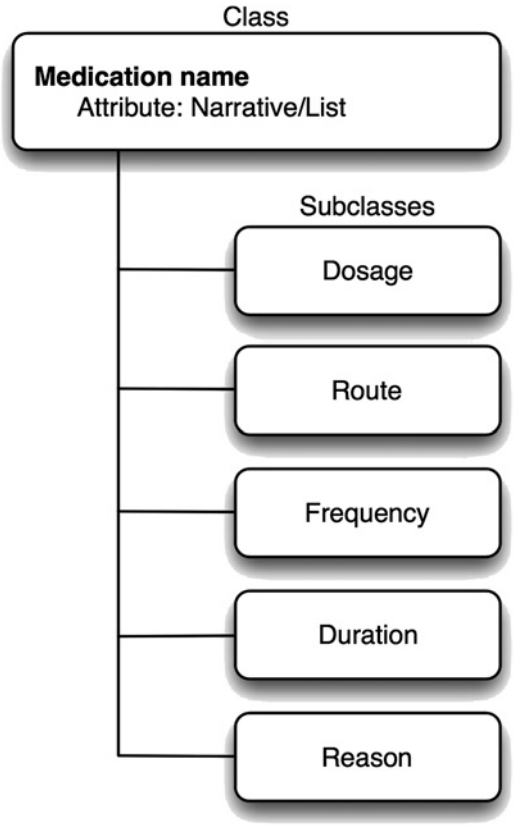

Figure 1 Medications and details annotation schema.

In order to estimate the consistency (reliability) of annotations created by our team, 10 documents were annotated by two team members. Agreement was measured using the inter-annotator agreement (IAA=matches/(matches+non-matches))[10 11] metric.

Textractor system description

Overview of the analysis pipeline

The architecture of Textractor is based on the UIMA framework, which enables definition and customization of the system through several descriptor files in XML format. The processing pipeline of our system is presented in figure 2. Each clinical document is loaded in the pipeline, then processed by several modules using either machine learning algorithms or pattern matching techniques (regular expressions, rules, dictionaries). The analysis starts with the document structure (detection of sections, of sentences, of tokens, and part-of-speech (POS) tagging). The disambiguation of common ambiguous abbreviations, and the MMTx-based extraction of medications and reasons for their prescription follow. Context analysis is the next step, followed by medication information extraction (extraction of dosage, route, frequency, and duration information). Finally, medication entries are constructed by combining medication names with the corresponding information, and by combining multiple entries for the same medication.

The following sections explain each processing step, and more details are available in the online-only appendix available at http://jamia.bmj.com.

Section detection

Section headers are first detected using regular expressions based on capitalization patterns, numbering, punctuation, and presence of carriage returns. A list of common false-positive headers is then used to filter the titles detected with the regular

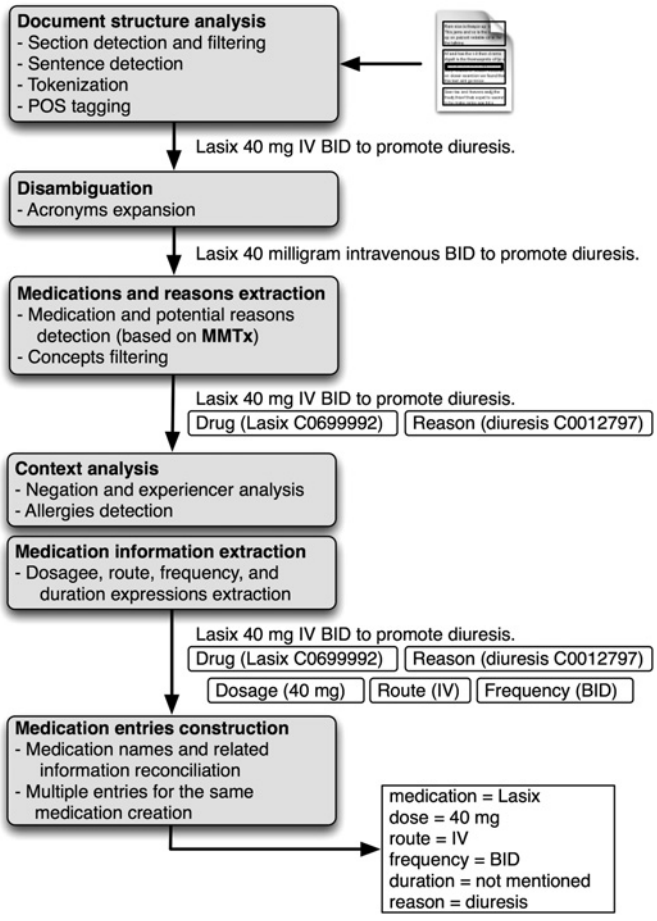

Figure 2 Medication name and details extraction process. BID, twice a day; IV, intravenous; MMTx, MetaMap Transfer; POS, part-of-speech.

expressions. The document structure is eventually represented as a tree with a maximum depth of three, with a header or subheader at each node. This tree representation is useful to build section annotations that store information on the parent sections (if any). Finally, a filter is used to exclude the sections that are likely to contain medications not taken by the patient (eg, 'Family History') or that had to be ignored for this challenge (eg, 'Allergies', 'Lab Results').

Sentence detection

We developed a UIMA wrapper similar to the one available in cTAKES[12] to integrate the OpenNLP sentence detector in our pipeline, and trained our own model to consider carriage returns as potential end-of-sentence markers. In cases where the length of the output sentence exceeds a certain number of characters, we proceed to further sentence splitting using regular expressions.

Tokenization and part-of-speech tagging

Because of the simple definition of the i2b2 tokens, a single regular expression is used to split the text into tokens. The part-of-speech tagger is based on an OpenNLP module, integrated into the pipeline with a UIMA wrapper. Each token is fed to the tagger and processed using the model provided with cTAKES (postagger.model.cTAKES1.bin.gz file).

Medications and reasons for their prescription extraction

We use MMTx to detect UMLS Metathesaurus concepts corresponding to medications and possible reasons for their prescription (eg, diseases, symptoms). Textractor implements

the MMTxAPILite class from MMTX 2.4.C and uses the default MMTx datasets (complete 2006 UMLS Metathesaurus) and search parameters, but with only selected semantic types. Nine semantic types are used to extract medications, and six semantic types to extract possible reasons for prescription (table 1).

We developed a UIMA wrapper to integrate MMTx in our pipeline. It takes a sentence as input and produces a list of UMLS Metathesaurus concepts mapped in the sentence as output. As mentioned above, MMTx was developed to analyze biomedical text (ie, scientific publications), and not clinical text (ie, EHR documents), which is the target here. This causes MMTx to misunderstand common acronyms and abbreviations such as: 'Dr.', detected as 'Diabetic Retinopathy' instead of 'Doctor.' To avoid these errors, a list of 60 acronyms and abbreviations with their corresponding long form is used to replace each ambiguous abbreviation found in the sentence passed to MMTx. This list comes from the automated problem list (APL) system developed by the first author[13] and was built manually.

Context analysis

Since MMTx provides no context analysis (eg, 'insulin' would be extracted from '…glucose management didn't require any insulin'), we developed a context analysis module based on a limited version of ConText[14] that infers possible negation and the experiencer. We enriched the original list of context analysis terms, and implemented the algorithm with a flexible window to analyze the context of concepts (instead of the original fixed five-word window).

Extraction of medication information

Dose, frequency, duration, and route phrases are detected independently using a set of regular expressions that were developed by analyzing a dozen documents from the corpus.

Reconciliation of medications and related information

The last step combines all these independent annotations to create medication entries. The algorithm implemented looks for medication information preceding or following the medication names detected by MMTx in the same sentence. The algorithm uses a variable window before and after the medication name, and a list of all extracted medication information elements. Each element that is linked to the medication name is removed from this list, to prevent different medications from being linked to the same information element.

Medication reasons, after extraction with MMTx and context analysis, are often preceded or followed by expressions such as 'because of …', 'due to …', '… was treated with.' A set of regular expressions was developed to detect these expressions. For the medications that were not reconciled with any reason after this first step, we enhance the extraction with a knowledge base of about 750 paired medications and diseases, built from the Pharmacogenomics Knowledge Base (PharmGKB[15]), the Comparative Toxicogenomics Database (CTD[16]), and the UMLS Metathesaurus, and explained in the online-only appendix available at http://jamia.bmj.com. If a possible reason is found in a window of two lines before or after 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 for the prescription.

Finally, multiple medication entries are created for a single medication name if the related information includes multiple values of the same category.

Evaluation metrics

Measurements for this challenge were performed at multiple levels. Exact matches and inexact matches were considered at the level of systems or at the level of documents, horizontally or vertically, as explained in Uzuner *et al*.[6]

Table 1 Unified Medical Language System (UMLS) semantic types used with MetaMap Transfer (MMTx)

Semantic type name	MMTx abbreviation	Use	Concept examples
Amino acid, peptide, or protein	aapp	M	Insulin lente, Zestril
Antibiotic	antb	M	Ofloxacin, vancomycin
Biologically active substance	bacs	M	Heparin, protamine
Carbohydrate	carb	M	Glucose, Lovenox
Hormone	horm	M	Levothyroxine, insulin
Organic chemical	orch	M	Amiodarone, clonazepam
Pharmacologic substance	phsu	M	Advair diskus, mocinazole nitrate
Steroid	strd	M	Rhinocort, prednisone
Vitamin	vita	M	Multivitamin, calcitriol
Disease or syndrome	dsyn	R	Pneumonia, decubitus ulcer
Congenital abnormality	cgab	R	Congenital exomphalos
Finding	fndg	R	Ecchymosis, falls
Pathologic function	patf	R	Edema, knee joint effusion
Sign or symptom	sosy	R	Wheezing, heart murmur
Therapeutic or preventive procedure	topp	R	Revision procedure

M, medications; R, possible reasons for the prescription.

RESULTS

Clinical corpus annotation

Each participating team annotated a number of documents proportional to the size of the team. We received 40 discharge summaries to annotate. A subset of 10 documents annotated by two team members was used to assess task consistency. Inter-annotator agreement for the 10 documents used to evaluate task consistency was highest for medication names (86%; 92% partial match) and lowest for durations (16%; 29% partial matching). More details are available in table 3 in the online-only appendix available at http://jamia.bmj.com. Agreement of annotations from the 20 participating teams was estimated with the same metrics as used for systems evaluation. These methods and results are reported in Uzuner *et al*.[6]

Official evaluation

We ran three slightly different versions of Textractor for this challenge. Each version used a different set of UMLS semantic types for MMTx. Version 1 used all 15 semantic types listed in table 1 (aapp, antb, bacs, carb, horm, orch, phsu, strd, vita and dsyn, cgab, fndg, patf, sosy, topp). Version 2 used a more limited set of semantic types for potential prescription reasons extraction (dsyn, patf, sosy). Version 3 was based on a small set of semantic types for both medications extraction and potential

Table 2 Results of the exact and inexact match evaluation

	Exact match			Inexact match		
Fields	Recall	Precision	F_1-measure	Recall	Precision	F_1-measure
Medication	0.746	0.772	0.759	0.763	0.784	0.773
Dose	0.757	0.916	0.829	0.786	0.925	0.850
Route	0.817	0.920	0.865	0.803	0.926	0.860
Frequency	0.789	0.892	0.837	0.742	0.924	0.823
Duration	0.326	0.397	0.358	0.326	0.501	0.395
Reason	0.169	0.669	0.270	0.148	0.703	0.245
Overall	0.715	0.832	0.769	0.693	0.839	0.759

prescription reasons extraction (antb, phsu, vita and dsyn, patf, sosy). Version 1 gave the best results, as listed in table 2, and the analysis of the 251 documents in the reference standard took this version of Textractor an average of 24 s per document. Most of the time was spent extracting concepts with MMTx, and even when using only a few semantic types and skipping sections of the document, the concept extraction phase still represented most of the execution time.

DISCUSSION

Results discussion

The official evaluation of Textractor for the i2b2 medication extraction challenge demonstrated satisfactory performance. Our system was among the 10 best performing systems for this challenge,[6] with a system-level overall F_1-measure reaching about 77% for exact matches, a recall of 72%, and a precision of 83%. Performance was the best with route (mode) information (F_1-measure about 86%), and was good for medication information such as dosage and frequency, with F_1-measures of about 82–85%. Results were not as good for durations, with F_1-measures of 36–39%, and for reasons, with F_1-measures of 24–27%. More results are available in the online-only appendix and in Uzuner *et al*.[6]

Textractor errors analysis

The poor recall for duration attributes (around 32%) can be explained by the fact that, on average, only 34% of these attributes detected through regular expressions were eventually linked with a medication. Multiple durations linked with the same medication were often missed. The low recall was also due to the insufficient coverage of the diversity of duration expressions in our small set of manual annotations, and to the same sentence scope limit for medication and attributes linking.

Reasons for a prescription were detected with a recall of 17% (exact matches), therefore more than 80% were not properly linked with a medication or were missed by our system. The former was by far the main issue. Here also, the rules and regular expressions used to link reasons with medications were insufficient, being limited to the same sentence. This limited scope was an important problem since reasons for medications can be listed in multiple places and may even be found outside the scope of text defined for this challenge. On average, about 2% of the potential reasons detected by MMTx were linked with a medication. Multiple reasons linked with the same medication were also often missed. The second method to reconcile reasons with medications, based on the drug–disease knowledge base, tries to overcome the scope limitation by extending the search to two lines before or after the medication name. Unfortunately, the drug–disease knowledge base was clearly insufficient and did not offer the various levels of granularity needed to match all reasons and medications detected by MMTx. The number of reason false negatives therefore mostly reflects the number of reasons that were not mentioned in the same sentence as the medication.

CONCLUSION

This challenge was a great opportunity to begin the development of a new more robust and flexible NLP system based on a standard architecture: UIMA. Because of the short development time frame, we focused on the integration of existing components rather than the development of new ones. In the end, we built a complete NLP system in only two months, with only a single developer. We used MMTx for its good UMLS Metathesaurus concepts indexing. Finally, this challenge provided an opportunity to develop information extraction functionalities we will need in current and future research, and comparatively evaluate several methodologies for automated information extraction for the research infrastructure we are building at the University of Utah Health Sciences Center.

Acknowledgments We thank the i2b2 challenge team for the development of the training and testing corpora and for the excellent organization of this challenge. The Textractor system and succinct results of its evaluation have been presented at the workshop organized at the end of the 'i2b2 medication extraction challenge' in San Francisco, California, USA, in November 2009.

Funding JFH was supported by National Library of Medicine grant R21 LM009967. The project described was supported in part by the i2b2 initiative, Award Number U54LM008748 from the National Library of Medicine. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Library of Medicine or the National Institutes of Health.

Ethics approval This study was conducted with the approval of the Partners Healthcare, Boston, MA, USA.

Provenance and peer review Not commissioned; externally peer reviewed.

REFERENCES

1. **Barach P,** Small SD. Reporting and preventing medical mishaps: lessons from non-medical near miss reporting systems. *BMJ* 2000;**320**:759–63.
2. **Kohn L,** Corrigan J, Donaldson M. *To err is human: building a safer health system*. Washington, DC: National Academy Press, 2000.
3. **KLAS Enterprises.** *CPOE digest 2006*. Orem, Utah: KLAS Enterprises, 2006.
4. **Apache.** UIMA (Unstructured Information Management Architecture). 2008. http://incubator.apache.org/uima/ (accessed 8 May 2008).
5. **Meystre SM,** Savova GK, Kipper-Schuler KC, *et al*. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb Med Inform* 2008:128–44.
6. **Uzuner Ö,** Solti I, Cadag E. Extracting medication information from clinical text. *J Am Med Inform Assoc* 2010;**17**:514–18.
7. **Lindberg C.** The Unified Medical Language System (UMLS) of the National Library of Medicine. *J Am Med Rec Assoc* 1990;**61**:40–2.
8. **Ogren PV.** Knowtator. http://bionlp.sourceforge.net/Knowtator/.
9. **Musen MA,** Eriksson H, Gennari JH, *et al*. PROTEGE-II: a suite of tools for development of intelligent systems from reusable components. *Proc Annu Symp Comput Appl Med Care* 1994:1065.
10. **Roberts A,** Gaizauskas R, Hepple M, *et al*. The CLEF corpus: semantic annotation of clinical text. *AMIA Annu Symp Proc* 2007:625–9.
11. **South BR,** Shen S, Jones M, *et al*. Developing a manually annotated clinical document corpus to identify phenotypic information for inflammatory bowel disease. *BMC Bioinformatics* 2009;**10**:S12.
12. **Savova G,** Kipper-Schuler K, Buntrock J, *et al*. *UIMA-based clinical information extraction system*. Marrakech, Morocco: LREC, 2008.
13. **Meystre S,** Haug PJ. Natural language processing to extract medical problems from electronic clinical documents: performance evaluation. *J Biomed Inform* 2006;**39**:589–99.
14. **Chapman WW,** Chu D, Dowling JN. ConText: an algorithm for identifying contextual features from clinical text. *Proceedings of the workshop on BioNLP 2007: biological, translational, and clinical language processing*. Prague 2007:81–8.
15. **The Pharmacogenomics Knowledge Base (PharmGKB).** http://www.pharmgkb.org/, 2010.
16. CTD: the comparative toxicogenomics database. http://ctd.mdibl.org/, 2010.