Automated Transformation of Semi-Structured Text Elements

Johannes Heurix
Research, Vienna, Austria., jheurix@sba-research.org

Antonio Rella
Xitrust Secure Technologies, Graz, Austria., antonio.rella@xitrust.com

Stefan Fenz
ISIS, Vienna University of Technology, Vienna, Austria., stefan.fenz@tuwien.ac.at

Thomas Neubauer
ISIS, Vienna University of Technology, Vienna, Austria., thomas.neubauer@tuwien.ac.at

Automated Transformation of Semi-Structured Text Elements

Johannes Heurix
SBA Research
jheurix@sba-research.org

Antonio Rella
XiTrust Secure Technologies
antonio.rella@xitrust.com

Stefan Fenz
Vienna University of Technology
stefan.fenz@tuwien.ac.at

Thomas Neubauer
Vienna University of Technology
thomas.neubauer@tuwien.ac.at

ABSTRACT

Interconnected systems, such as electronic health records (EHR), considerably improved the handling and processing of health information while keeping the costs at a controlled level. Since the EHR virtually stores all data in digitized form, personal medical documents are easily and swiftly available when needed. However, multiple formats and differences in the health documents managed by various health care providers severely reduce the efficiency of the data sharing process. This paper presents a rule-based transformation system that converts semi-structured (annotated) text into standardized formats, such as HL7 CDA. It identifies relevant information in the input document by analyzing its structure as well as its content and inserts the required elements into corresponding reusable CDA templates, where the templates are selected according to the CDA document type-specific requirements.

Keywords

Information Extraction, Text Transformation, Clinical Document Architecture, Semi-Structured Text

INTRODUCTION

Health care is no longer solely tasked with simply curing sick people. The increase in knowledge, advancements in medical techniques and the overall quality of diagnostics have led to a massive increase in information processing and storage demands. Managing the vast amounts of information produced nowadays is one of the main challenges of today's health care. As in virtually all domains these days, information and communication technologies (ICTs) have found its way into health care to manage the ever growing quantities of data. E-health denotes the application of ICTs to support medical workflows and to improve the communication between health care providers. Interconnected systems, such as electronic health records (EHR), are aimed to enhance the data processing facilities while keeping the costs at a controlled level (Chaundry, Wang, Wu, Maglione, Mojica, Roth, Morton, and Shekelle, 2008). Initiatives like 'Integrating the Healthcare Enterprises' (Integrating the Healthcare Enterprise (IHE), 2009) have published communication standards to harmonize data exchanging and sharing mechanisms to ultimately improve patient care.

However, truth is that although specialized medical data processing systems, especially image processing systems, are highly advanced and effective, organizational and administrative systems lack well behind their potential capabilities. Today's ICT environment in medical facilities is characterized by numerous legacy systems and isolated applications including hospital information systems. Textual information is stored in different data formats, which complicates data sharing between different systems. Furthermore, many digitized documents, such as discharge letters, clinical notes, or medical histories, are in narrative form making automated data processing much more difficult. The XML-based Health Level 7 Clinical Documents Architecture (HL7 CDA (Health Level Seven Inc., 2007)) provides a common standard for representing medical information in a structured way. CDA documents separate administrative information and the actual medical content into header and body sections. This separation between administrative and medical sections facilitates privacy-preserving storage and data sharing mechanisms, such as pseudonymization (cf. Neubauer and Heurix, 2011). The difficulty in creating CDA

documents is to extract the relevant information from the unstructured documents of different data formats of the legacy systems, to convert it, and to insert it into the corresponding sections of CDA.

This article presents a simple yet effective rule-based transformation system to automatically convert semi-structured narrative medical texts (i.e., annotated input strings) into CDA-conforming documents by identifying relevant information and inserting it into composable CDA templates. Both transformation rules and CDA templates are structured in XML and are separated from each other in order to support different document type-specific sets of rules to be applied to the same templates, depending on the input string's semantic structure. The rules' syntax allows non-technical domain experts with minimum XML knowledge to design complex extraction conditions, while the separation of the rules from the CDA templates also facilitates reusability. Due to its modularity, the system can be combined with different annotation engines, such as the GATE framework (cf. Cunningham, Maynard, Bontcheva, Tablan, Aswani, Roberts, Gorrell, Funk, Roberts, Damljanovic, Heitz, Greenwood, Saggion, Petrak, Li, and Peters, 2011).

BACKGROUND

Information extraction (IE) is a sub-domain of the natural language processing (NLP) discipline and is tasked with the automated extraction of structured information from unstructured sources (Sarawagi, 2007). A typical IE system has two objectives: (i) identifying and annotating potentially relevant information in the narrative input text and (ii) the actual extraction and transformation of the desired information into the target form. The annotation task usually involves multiple pre-processing steps including tokenization, part-of-speech tagging, or parsers for boundary and named-entity recognition which are executed in a pipeline where the output of the former step is used as input for the next step to improve the quality of the overall result. Existing work largely does not distinguish entity recognition or annotation and actual information extraction. Text processing steps are composed into adaptable annotation frameworks, such as GATE (Cunningham et al., 2011), C-PANKOW (Cimiano, Ladwig, and Staab, 2005) or UIMA (Ferrucci and Lally, 2004) which provide their different processing functionality, such as tokenization or whitespace identification through special plug-ins.

IE systems can be categorized into two fundamentally different types (Appelt and Israel, 1999): (i) the Knowledge Engineering Approach with hand-made rules written by domain experts to identify and correctly mark the relevant information entities, and (ii) the Automatic Training Approach where the system creates the rules itself by analyzing manually pre-labeled (annotated) training corpora. There has been an ongoing debate on which of these approaches performs better (Appelt and Israel, 1999; Sarawagi, 2007): Knowledge engineering-based systems tend to produce good results very fast, especially when training data is tedious and costly to acquire. They also excel when the annotation and extraction specifications are likely to change in a foreseeable way (e.g., when the layout of documents are updated). The big downside of hand-crafted rules is the actual work to define them. Creating accurate rules often relies on a tedious and iterative testing and adapting process. Training-based systems relieve the domain expert from this manual rule-creation work which means that no (potentially) complex rule formalisms need to be learned. But in essence, automatic training-based systems shift the workload from creating the rules to annotating the training corpora. To produce reasonably good results, these systems require a large number of manually annotated documents. Thus, to select the adequate system for a particular IE problem, it highly depends on the availability of training data and their structure, the technical expertise of domain experts, as well as the structural and lexical properties of the document base.

IE systems have been applied in a multitude of applications including enterprise applications, scientific purposes, or web-based systems (Sarawagi, 2007). Although originally developed outside the biomedical and clinical area, information extraction has since been adapted to the medical domain as well. One of the first systems developed and successfully used was the often cited MedLEE system (Friedman, Alderson, Austin, Cimino, and Johnson, 1994; Friedman, Johnson, Forman, and Starren, 1995), an NLP system to identify radiology information in narrative text to be mapped to a structured representation containing clinical terms. Some other work dealing with radiology reports include (Haug, Ranum, and Frederick, 1990) or more recently (Friedlin and McDonald, 2006). Information extraction methods were also applied in the clinical domain in other contexts including discharge summaries, e.g., (Sibanda, He, Szolovits, and Uzuner, 2006), (Long, 2005), and (Zeng, Goryachev, Weiss, Sordo, Murphy, and Lazarus, 2006), either for extracting generic information or for special purpose, such as extracting ICD codes (Zweigenbaum, Bachimont, Bouaud, Charlet, and Boisvieux, 1995). Another popular application area of information extraction in the clinical context involves the automated de-identification of medical documents for secondary use. Exemplary work include the identification of names only (Taira, Bui, and Kangarloo, 2002), systems based on GATE (Guo, Gaizauskas, Roberts, Demetriou, and Hepple, 2006) or MedLEE (Morrison, Li, Lai, and Hripcsak, 2009), or systems specifically tailored for a particular language, such as French (Grouin, Rosier, Dameron, and Zweigenbaum, 2009) or Swedish (Velupillai, Dalisanisa, Hassela, and Nilsson, 2007).

Unlike these approaches, this work focuses on the automated transformation of medical documents into the HL7 (Version 3) Clinical Document Architecture (Health Level Seven Inc., 2007) format, although the system can be easily adapted to transform information to any XML-based document format. The transformation process includes the automated extraction of relevant text passages and entities and the insertion into CDA-conforming XML documents. CDA documents are organized into header sections, containing administrative data, such as the patient's name and address encapsulated into mandatory XML blocks, and more flexible body sections with the actual medical data. Depending on the CDA level (1 to 3), this content is organized in raising granularity: While level 1 contains largely free narrative text blocks only, in level 3 documents the narrative blocks are extended with specially encoded machine-readable sections. Codes are taken from established standards from the clinical domain, such as the Logical Observation Identifiers Names and Codes (Regenstrief Institute, Inc., 2008) and International Statistical Classification of Diseases and Related Health Problems (ICD Version 10 (World Health Organization, 2007)).

Considering a typical discharge letter, we identified the following requirements for a document transformation system (examples are given in the brackets):

- Identification and extraction of single named-entities (names, locations).
- Distinction between different instances of the same entity class (patient name vs. health professional name).
- Composition of entities (social security number with birth date).
- Limited sections of text blocks (ICD codes).
- Complete text blocks (diagnosis).
- Multiple successive text blocks (medical history extending over multiple paragraphs).

TRANSFORMATION OF SEMI-STRUCTURED TEXTUAL INFORMATION

In this paper, we present a rule-based transformation methodology for converting medical information extracted from semi-structured texts into CDA-conforming documents with the following characteristics:

- Using a rule-based approach to exploit the similarity of medical documents (e.g., discharge letters almost always contain sections including diagnosis, reason for visit, procedures, etc. separated into logical units, such as paragraphs) making writing rules easier and, thus, achieving results faster without a large training set.
- Separating annotation from actual transformation to benefit from different annotation engines, such as GATE, C-PANKOW, or UIMA.
- Separating transformation rules from target document templates to facilitate template reuse when adapting rules to a different document set (e.g., different layouts of medical records depending on the creation date).
- Designing the transformation rules' syntax and semantics to allow non-technical domain specialists the development of complex rules covering all requirements without learning complex formalisms (e.g., XSLT combined with complex XPath expressions).
- Using both structural and content-related knowledge to formulate the rules.

The transformation methodology relies on the annotation engines to provide annotated semi-structured input strings. As a precondition, the string needs (i) to be partitioned into semantically-contained text blocks, such as paragraphs (section boundaries identification) and (ii) annotated with pre-specified entity classes, such as names or dates (named-entity recognition). Our prototype accepts input strings structured as XML documents where the text blocks are surrounded with <p> tags. Each text block in turn can have multiple named-entities, again annotated with XML tags (e.g., <a_firstname>). The document templates are organized into logical CDA sections (recordTarget, author, diagnosis, etc.) and contain the static text body (predefined) and references where the extracted information should be inserted into. Templates are selected and composed depending on the document types (discharge letter, lab results, etc.). Each of the templates is assigned one or more rules which are expressed in XML as well and contain sub-rules for each entity-of-interest in each template. Multiple rules for a single template account for different document subtypes (e.g., layout changes of discharge letter in the course of time).

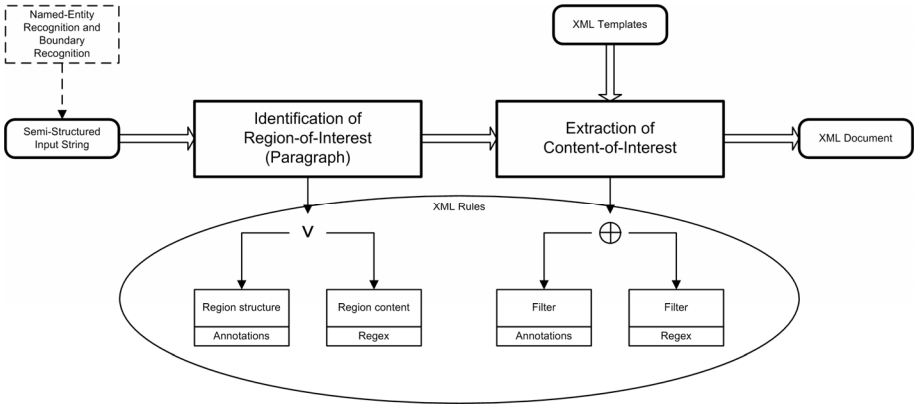

Figure 1: Transformation Process

The transformation process is depicted in Figure 1:

1. Identification of the document (sub)type and selection of the appropriate rules and templates.
2. For each template:
    a. Identification of the region-of-interest (one or more paragraphs) using region structure (combination of annotations) and/or content (regular expression) information as decision tools.
    b. Extraction of the actual content-of-interest from the region-of-interest using either region structure or content information as filters.
    c. Insertion of the content-of-interest into the corresponding sections in the template.
    d. Continuation with the next template until all templates and corresponding rules are processed.
3. Composition of the templates into a full CDA document.

In the following, templates and rules, their semantics, and text processing effects are described in detail. As the syntax is XML-based, both templates and rules must conform to XML schema definitions. For better overview, the template and rule structures are represented as figures with boxes as XML elements (nodes), where element attributes are shown below and element text content on the right. Parent/child relationships of the nodes are expressed by connecting lines with cardinality indicators, extended with sequence/choice bars if applicable.

Templates

A template represents a building block (e.g., diagnostic section) that needs to be filled with the extracted information and combined with other templates to form the complete CDA document. Each <template> (see Figure 2) has a unique 'ID' and contains the <itemList> and <content> sections. While <content> contains the static text building blocks with empty sections that need to be filled with information extracted from the input string, <itemList> contains a set of <itemPath> nodes corresponding to each empty field in the <content> section. An <itemPath> definition has an ID, optional prefixes and suffixes (i.e., static text that is added to the extracted information like area codes), and an indicator whether the particular field is optional or not (e.g., patient names are mandatory while telecom values are not). The XPath expression specifies the field's location in the <content> section.

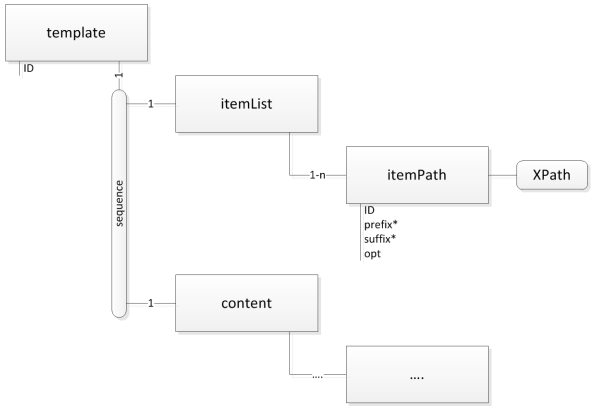

Figure 2: Template Structure

Rules

A rule encodes the information of how to identify (region-of-interest) and extract the relevant entities (content-of-interest) from the source input string. Each <rule> (see Figure 3) must correspond to a particular template indicated by the 'targetTemplate' attribute. The other attribute defines a document subtype for which the rule is applicable (e.g., discharge letters from different wards within a hospital). Each rule contains one or more <region> nodes representing the regions-of-interest (i.e., paragraphs) where the <conditions> section denotes the structural and content-related conditions for finding the correct region and <contentMapping> the actual content that needs to be extracted from the region and inserted into the corresponding parts of the CDA templates.

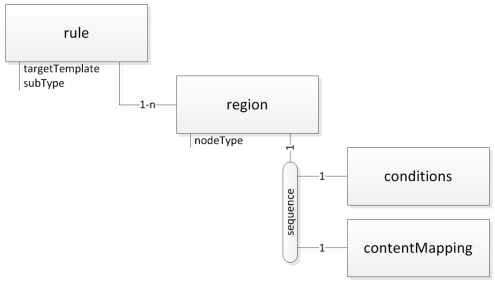

Figure 3: Rule Structure

The region's 'nodeType' attribute defines different ways of how to find the particular nodes of the regions-of-interest. Depending on 'nodeType', a different set of child <(begin/end)nodeCondition> elements are necessary in the <conditions> section (see Figure 4). Types, required children in the <conditions> section, and purposes are as follows:

- The 'single' type states that the region consists of a single paragraph only and, thus, requires only a single <nodeCondition> element. It is used to get a node where its composition is known relatively well.
- The 'singleSelected' type states that the region consists of multiple paragraphs that do not necessarily occur in immediate succession in the source input string. Only a single node per node condition is returned. It requires two or more <nodeCondition> elements and is used to get multiple nodes where each node condition refers to a single node and the nodes' compositions are known relatively well.
- The 'multiSelected' type states that the region consists of all nodes matching any node condition and requires one or more <nodeCondition> elements. It is used to get multiple nodes where the nodes may contain any known regular expression keywords and/or tags (see below).
- The 'multiContinuous' type states that the region contains all nodes between a beginning node, <beginNodeCondition>, and an ending node, <endNodeCondition>. The 'conditionType' attribute determines whether the begin/end nodes are included in the region or not. The 'multiContinuous' type is used to get all nodes within two known boundaries.

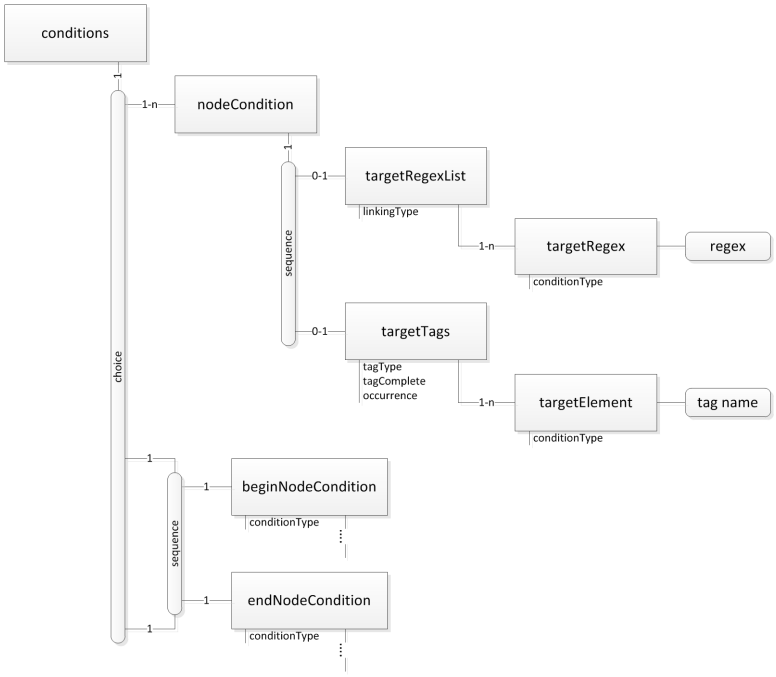

Figure 4: Rule – Conditions Structure

The <nodeCondition> element contains either a <targetRegexList> or a <targetTags> element, or both. The former describes a condition on the paragraph's content, while the latter states the set of required XML tags, i.e., annotations within the paragraph.

The <targetRegexList> contains one or more <targetRegex> elements with the actual regular expression as text value. The attributes 'linkingType' and 'conditionType' indicate how the individual regular expressions are linked (and/or) and whether the particular regular expression must or must not yield a match in the paragraph's text in order to be part of the region-of-interest (include/exclude). Similarly, <targetTags> contains one or more annotations defined as <targetElement> and tag name as text values which have to be occurring either in a particular sequence or in any sequence (tagType). The 'tagComplete' attribute defines if the paragraph must not contain any additional annotation other than those of <targetElement>, while 'occurrence' states the desired occurrence of the annotation combination (e.g., occurrence = 2 matches the second paragraph with the particular annotation combination).

Apart from the 'conditionType' as attributes (explained further above), <beginNodeCondition> and <endNodeCondition> are composed just like the <nodeCondition> elements.

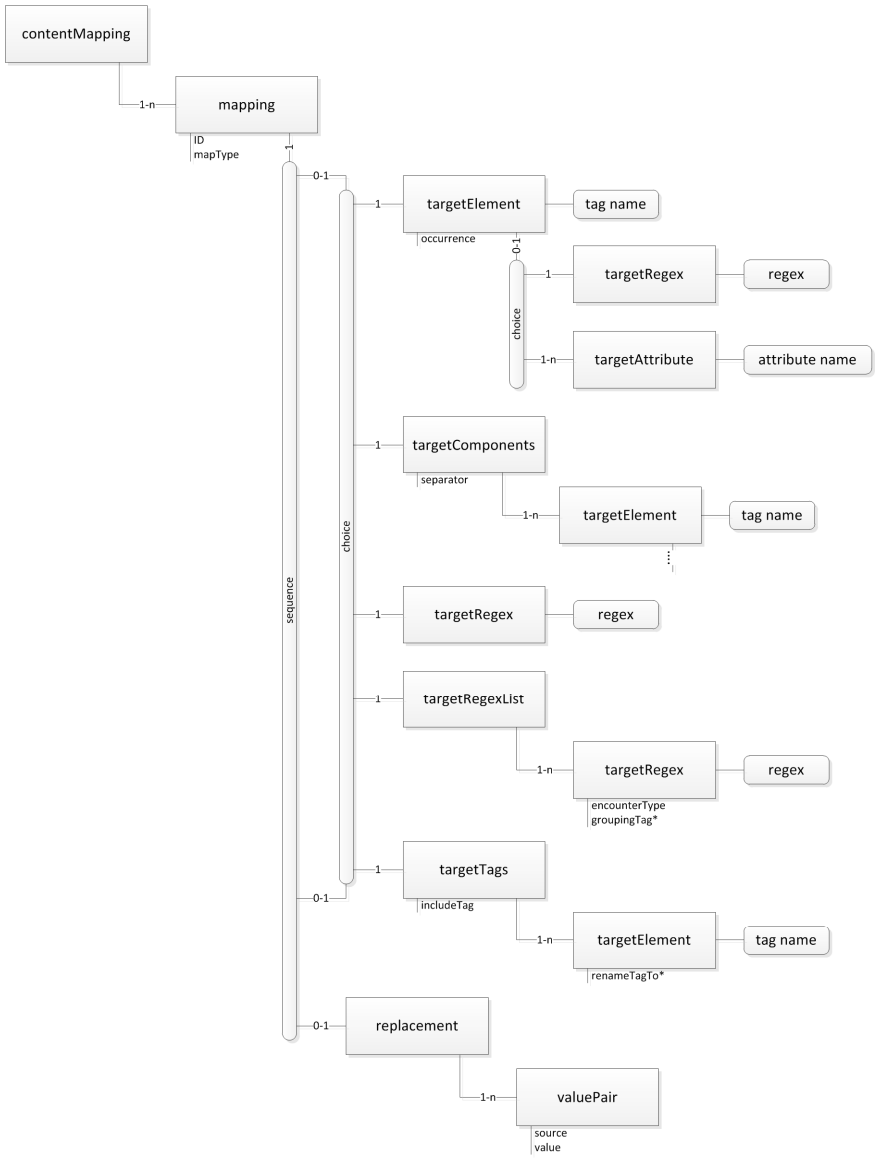

Figure 5: Rule - Content Mapping Structure

While <conditions> indicate the conditions under which the paragraphs belong to the region-of-interest of the current rule, <contentMapping> defines what information to actually extract from the region-of-interest. The section contains one or more <mapping> elements. The attribute 'ID' indicates where to insert the extracted information into the templates (matching the <itemPath> 'ID' attribute), while 'mapType' defines how to extract the information as follows:

- The 'singleElement' type represents a 1:1 mapping of an annotation tag's content (determined by tag name and its 'occurrence' within the region-of-interest, e.g., second occurrence of <a_firstName> as the patient's first name) to a particular section within the CDA template. Optionally, the content can be filtered by a regular expression or be composed of the annotation tag's attributes instead of the text content (indicated by one or more <targetAttribute> elements).
- The 'composedElement' type represents a content written into the CDA template composed of multiple annotated <targetElement> tags, separated by a 'separator' (default is an empty string), e.g. the composition of the Austrian 10-digit social security number using the 4-digit version combined with the person's birth date. The <targetElement> elements again can be filtered by regular expressions or attribute values.

- The 'fullContent' type simply refers to extracting the whole content of the region-of-interest into the corresponding section of the CDA template. It can (optionally) be filtered by a single <targetRegex>. An example for 'fullContent' is to copy the complete narrative paragraph of a diagnosis to the CDA template, while it may be filtered to extract only the ICD codes.
- The 'multiContent' type refers to multiple elements within the region-of-interest and inserted into the CDA template separated by a default grouping tag[1], unless otherwise specified. The content can be filtered either by a <targetRegexList> or by <targetTags>. The <targetRegexList> contains a set of one or more <targetRegex> elements with attributes 'encounterType' and 'groupingTag'. The former attribute distinguishes between extracting either only the first regular expression-matching string or all matching strings within the region-of-interest, while the optional latter attribute overrides the default grouping tag with an alternative one for each individual regular expression. <targetTags> contains the set of <targetElement> tags whose contents are to be extracted. The 'includeTag' attribute indicates whether the elements' original tags should be copied to the template too (in this case, the default grouping tag is replaced with the original one). The optional attribute 'renameTagTo' allows to individually rename the tag to an arbitrary one for each <targetElement>. Both <targetTags> and <targetRegexList> have their uses, e.g., to extract ICD codes from a diagnostic text spanning over multiple paragraphs, depending on how these are annotated by the annotation engine: If they are already annotated, <targetTags> can be used to extract them with potentially renaming the annotation tags with arbitrary ones matching the CDA standard. If the annotation is not able to annotate them, <targetRegexList> may contain the regular expressions to extract them (in this case with encounterType = all), again with an optional renamed grouping tag.

All mapping sections can have an additional and optional <replacement> section which contain <valuePair> elements. These refer to text elements ('source') that should be replaced with another text ('value'), e.g., if the word 'female' is encountered within a <gender> tag, it should be replaced with 'F' before being inserted into the patient template of the CDA header section.

CASE STUDY AND EVALUATION

We have developed a prototype written in Java as part of a system to automatically convert archived and digitized paper-based discharge summaries into the CDA document standard to be used as research knowledge base. Apart from the transformation module, the system consists of a Tesseract-based OCR (Optical Character Recognition) module[2] for converting scanned images of the discharge summaries to machine-readable text strings and an annotation module based on GATE[3] to extend the simple text string with annotations of PHI elements, i.e., Personal Health Information including names, addresses, and other patient-identifying information (U.S. Department of Health & Human Services Office for Civil Rights, 2003).

For the sake of this evaluation, we focus on examining the following CDA document segments:

- Header (normalized sections): recordTarget (patient's personal information), author (usually the treating physician), informationRecipient (physician receiving the discharge summary).
- Body: reason for visit (incidents, problems), diagnosis (results and outcome), procedures (medications and treatments), plan of care (recommended medications).

We investigated 30 documents from two different wards (internal medicine and surgery) of the same hospital. Although from the same hospital, the layouts of the documents differed in multiple aspects. This resulted in the need for different rule sets for each layout. These rule sets only required localized changes to factor in, e.g., modified patient blocks. Thus, the majority of the rule elements could be reused.

[1] As this is only used in CDA body templates, we simply use <paragraph> here.

[2] http://code.google.com/p/tesseract-ocr/

[3] http://gate.ac.uk/

	Total blocks	Total errors	OCR errors	Annotation errors	Transformation errors	Not in document
recordTarget	30	12	5	6	1	0
author	30	11	8	2	1	0
informationRecipient	30	16	0	4	1	11
reason	30	1	1	0	0	0
diagnosis	30	0	0	0	0	0
procedures	30	3	1	0	2	0
plan	30	6	0	0	2	4

Table 1: Evaluation Results

Table 1 illustrates the evaluation results. As the main performance indicator of the transformation process is how well the regions-of-interest are identified, we opted to not evaluate the f-measure for all entity classes (first name, last name, house number, etc.) individually. Instead, we focused on tracking the number of incomplete or erroneous CDA document blocks (segments), i.e., if a particular block missed a single entity, it was regarded as erroneous due to incorrectly identified regions-of-interest. The evaluation showed a significantly higher percentage of total errors in the CDA header blocks (40%, 37%, 53%) than in the CDA body blocks (3%, 0%, 10%, 20%), although at a different 'granularity'. Usually body blocks were filled by simply copying whole paragraphs (regions-of-interest) into the corresponding sections, thus, errors resulted in empty templates. CDA header sections, however, required the extraction of particular named-entities only, which means that the logged errors were the result of a single or two missing elements, such as a last name or street name in an address block.

An in-depth analysis of the errors revealed the causes: As can be seen in Table 1, we classified the errors into OCR (errors due to incorrectly identified characters and thus misspelled words or incorrectly identified text blocks), annotation (incorrectly annotated or missing named-entities), and actual errors due to limitations of the transformation system. Furthermore, some elements were simply missing in the source documents (informationRecipient, plan of care). The predominant OCR errors in the recordTarget and author sections were the result of misspelled first/last names (interfering written signatures and greyed background boxes in the source document), which were in turn not annotated and, thus, not identified by the transformation system. While the CDA header rules were largely composed of tag-specific conditions, the CDA body blocks relied on static known keywords (e.g., 'Current Status'), which facilitated the correct identification of those regions-of-interest.

The analysis of the remaining errors in the CDA body sections revealed the limitations of the transformation system: The two missing entries for procedures and plan occurred due to an incomplete keyword set. Other problems occurred when handling unexpected changes, such as keywords ('to be sent to') left out in the source document or structural variations (document author cited at the beginning of the document instead of at the end). In order to fill in the gaps (information missing in the source documents, such as hospital name) and verify the results (especially for the named-entities in the header section), we extended the transformation system to use additional information sources, such as archive metadata, and also created explicit rule sections for these additional sources.

In general, considering only the annotated document as information source, the better the quality of the OCR and annotation results, the better is the performance of the transformation system. The option to use either or both the source document's structure (annotation tags) and content (regular expression) to identify the intended regions-of-interest allows to adapt the transformation system to different annotation engines with varying annotation quality (e.g., compensating for missing annotation classes).

CONCLUSION

The differences in todays' health document formats considerably reduce the efficiency of data sharing. The HL7 CDA provides a standardized framework to represent multiple document types including radiology results or discharge summaries. This work presented a rule-based transformation system to automatically convert semi-structured source documents into the CDA standard. It is designed to (i) focus on the actual transformation process to be combinable with different annotation engines, (ii) facilitate the rule creation process for non-technical domain specialists without learning complex formalisms, and (iii) make use of both structural and content-related knowledge to formulate the rules. Further work includes the extension of the rule system to allow for more powerful rule conditions, the introduction of pre-specified referenceable utility-functions to further simplify the rule specifications, and the evaluation with different annotation engines.

ACKNOWLEDGMENTS

This research was funded by BRIDGE (#824884) and by COMET K1, FFG – Austrian Research Promotion Agency.

REFERENCES

1. Appelt, D. E., and Israel, D. J. (1999). Introduction to Information Extraction Technology. A Tutorial Prepared for IJCAI-99.
2. Chaundry, B., Wang, J., Wu, S., Maglione, M., Mojica, W., Roth, E., Morton, S. C., and Shekelle, P. G. (2008). Systematic Review: Impact of Health Information Technology on Quality, Efficiency, and Costs of Medical Care. *Annals of Internal Medicine*, *144*, 742-752.
3. Cimiano, P., Ladwig, G., and Staab, S. (2005). Gimme' the context: Context-driven automatic semantic annotation with C-PANKOW. *Proceedings of the 14th International Conference on World Wide Web*, 332-341.
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). Developing Language Processing Components with GATE Version 6 (a User Guide). University of Sheffield, Department of Computer Science.
5. Ferrucci, D., and Lally, A. (2004). Uima: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering, 10*, 327-348.
6. Friedlin, J., and McDonald, C. J. (2006). A Natural Language Processing System to Extract and Code Concepts Relating to Congestive Heart Failure from Chest Radiology Reports. *AMIA Annual Symposium Proceedings*, 269-273.
7. Friedman, C., Alderson, P. O., Austin, J. H., Cimino, J. J., and Johnson, S. B. (1994). A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association, 1*(2), 161-174.
8. Friedman, C., Johnson, S. B., Forman, B., and Starren, J. (1995). Architectural requirements for a multipurpose natural language processor in the clinical environment. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 347-351.
9. Grouin, C., Rosier, A., Dameron, O., and Zweigenbaum, P. (2009). Testing tactics to localize de-identification. *Medical Informatics in Europe conference (MIE'2009)*, 735-739.
10. Guo, Y., Gaizauskas, R., Roberts, I., Demetriou, G., and Hepple, M. (2006). Identifying personal health information using support vector machines. *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data.*
11. Haug, P. J., Ranum, D. L., & Frederick, P. R. (1990). Computerized extraction of coded findings from free-text radiologic reports. *Radiology, 174*, 543-548.
12. Health Level Seven Inc. (2007). HL7 Version 3. *Retrieved from http://www.hl7.org/v3ballot/html/welcome/environment/index.htm.*
13. Integrating the Healthcare Enterprise (IHE). (2009). IHE IT Infrastructure (ITI) Technical Framework 6.0.
14. Long, W. (2005). Extracting diagnoses from discharge summaries. *AMIA Annual Symposium Proceedings*, 470-474.
15. Morrison, F., Li, L., Lai, A., and Hripcsak, G. (2009). Repurposing the clinical record: Can an existing natural language processing system de-identify clinical notes? *Journal of the American Medical Informatics Association, 16*, 37-39.
16. Mystre, S., Savova, G. K., Kipper-Schuler, K. C., and Hurdle, J. F. (2008). Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research. *IMIA Yearbook of Medical Informatics*, 128-144.
17. Neubauer, T., and Heurix, J. (2011). A methodology for the pseudonymization of medical data. *International Journal of Medical Informatics, 80*(3), 190-204.
18. Regenstrief Institute, Inc. (2008). Logical Observation Identifiers Names and Codes (LOINC).
19. Sarawagi, S. (2007). Information Extraction. *Foundations and Trends in Databases*, 261-377.
20. Sibanda, T., He, T., Szolovits, P., and Uzuner, O. (2006). Syntactically-informed semantic category recognition in discharge summaries. *AMIA Annual Symposium Proceedings*, 714-718.

21. Taira, R. K., Bui, A. A., and Kangarloo, H. (2002). Identification of Patient Name References within Medical Documents Using Semantic Selectional Restrictions. *AMIA Annual Symposium Proceedings*, 757-761.

22. U.S. Department of Health & Human Services Office for Civil Rights. (2003). Summary of the HIPAA Privacy Rule. *Online*.

23. Velupillai, S., Dalisanisa, H., Hassela, M., and Nilsson, G. H. (2007). Developing a Standard for De-Identifying Electronic Patient Records written in Swedish: Precision, Recall and f-Measure in a Manual and Computerized Annotation Trial. *International Journal of Medical Informatics Association, 14*, 564-573.

24. World Health Organization. (2007). International Statistical Classification of Diseases and Related Health Problems (ICD).

25. Zeng, Q. T., Goryachev, S., Weiss, S., Sordo, M., Murphy, S. N., and Lazarus, R. (2006). Extracting principal diagnosis, co-morbidity and smoking status for asthma research: Evaluation of a natural language processing system. *BMC Medical Informatics & Decision Making, 6*(30), 30-39.

26. Zweigenbaum, P., Bachimont, B., Bouaud, J., Charlet, J., and Boisvieux, J. F. (1995). A multilingual architecture for building a normalised conceptual representation from medical language. *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, 357-361.