Information Extraction from Echocardiography Records

Georg Fette * and **Peter Kluegl** * and **Maximilian Ertl** † and **Stefan Störk** ‡ and **Frank Puppe** *

* University of Würzburg, Dep. of Computer Science VI, Würzburg, Germany, {fette, pkluegl, puppe}@informatik.uni-wuerzburg.de

† University Hospital Würzburg, Servicezentrum Medizininformatik, Würzburg, Germany, maximilian.ertl@smi.uni-wuerzburg.de

‡ University Hospital Würzburg, Medizinische Klinik und Poliklinik I, Würzburg, Germany, stoerk_s@klinik.uni-wuerzburg.de

Abstract

Electronic health records are a rich source for medical information. However, large parts of clinical diagnosis reports are in textual form and are therefore not per se usable for statistical evaluations. To transform the information from an unstructured into a structured form is the goal of medical language processing. In this paper we want to propose an approach for the creation of a training corpus for information extraction from echocardiography reports and the creation of a sequence labeler based on keyword matching and window-based disambiguation. The outcomes presented in this paper are the first results from ongoing work from a series of medical projects.

1 Introduction

The structured representation of information from medical reports is useful for tasks like clinical research based on statistics. Parts of electronic health records, like laboratory findings are already accessible in a structured form but significant portions are stored in plain text format which have to be transformed into a structured representation for further processing. Most clinical records consist of a mixture of narrative text and a great amount of medical abbreviations. Although for an inexperienced reader those kinds of reports seem quite inscrutable, it takes only little training to get accustomed to the sub-language used in the specific domain. Especially, procedure based domains like echocardiography records possess a quite manageable sublanguage, as they always deal with the same objects (e.g. in echocardiography: the heart) and therefore they include similar details. The scope of a domain is therefore an important aspect in information extraction (IE).

As a special feature of medical records there can be defined different types of free texts. A report can be created automatically by simple concatenation of already structured information like measured values. These unstructured texts often pose only little problems for converting them back into a structured form. Another text type involves the use of text building blocks with place holders. Many clinical institutions use systems to formalize the composition of reports by offering predefined text blocks. These components do not always cover the whole bandwidth of possible information so the preliminary composed texts are often edited manually. Furthermore, persons unfamiliar with the supporting text block system ignore big parts of the coded sections and rely on writing their reports mainly in narrative form. As these component systems were introduced only in recent years there is still much data which exists in a more complex, free form. This completely narrative form is the third and most complicated to process form of medical texts. They pose the biggest variety in the way information is expressed. All text forms are often blended into each other. For example some reports are composed of computer generated parts and text block sections and, additionally, the final documents undergo manual corrections.

The aim of this paper is to describe a general method for processing all kinds of medical records. In the current stage of our project we are working with document structures which are relatively easy to process and we will continue with more and more complex document types. We propose a work cycle for IE for cost-efficiently creating a training corpus for a supervised learning method. Furthermore, we present a fast algorithm for the automatic extraction of predefined information types from plain texts.

The desire for the automatic extracting of information from unstructured medical texts is not new and for decades enormous effort has been made to solve this problem [Hripcsak *et al.*, 1995]. The various IE systems range from the analysis of documents with a very diverse vocabulary like discharge summaries (e.g.MedLee [Friedman, 2000]) to more specific domains like radiography reports [Mamlin *et al.*, 2003] or, like in our case, echocardiography reports [Chung and Murphy, 2005] [Denny and Peterson, 2007]. Another group of systems focuses on the extraction of medical codes like ICD-10 or ICD-9 diagnosis codes. In this domain MedLee also has to be mentioned as a well performing system with high adaptability even for unknown domains. Further systems are MetaMap [Aronson, 2001] and cTAKES [Savova *et al.*, 2010]. The methods applied in the different systems range from simple pattern matching approaches to grammatical analysis like POS tagging and parsing of the texts. An exhaustive review on IE from textual documents in electronic health records can be found in [Meystre *et al.*, 2008].

2 Problem Analysis

We would like to extract information from texts so that the resulting data is a list of attribute-value pairs. Before IE it is necessary to define the different attribute types in the data. Each information type $t_i = (id_i, name_i)$ is defined by an identifier and/or name. We call the definition of the types $T = \{t_1, .., t_n\}$ a *terminology*. The extracted instances $a_i = (t_i, value_i, \text{offsets}_i)$ have to belong to one of the predefined types. We call a document $d_i = (text_i, \emptyset)$ without the included information an *unlabeled document*, one with its extracted information $d_i = (text_i, A_i), A_i = (a_1^i, ..a_n^i)$ a *labeled document*. The task of defining a proper terminology is part of the IE itself, as it is apriori unknown of what type the extracted information will be. When dealing with a

new domain it is difficult to define a complete terminology for all of the possible information types in the specific domain. Although parts of the terminology can be drawn from an expert's background knowledge, there exists a significant gap between the knowledge structure independently created from given data and the intrinsic unknown information structure. To avoid such a gap it is recommended to define the terminology with respect to the data analysis process. We create the terminology from the documents by analyzing random documents and adding types to the terminology whenever an unknown type is encountered. The creation of new terminology entries can go hand in hand with the manual labeling process.

There are two approaches for automatic IE from texts: supervised and unsupervised. In unsupervised IE the results are almost completely defined by the data itself. In supervised IE a human instructor can provide examples how the extraction should take place. As in the domain we want to work on, we already have background knowledge we want to use, we have decided to use the supervised IE approach. The common way for supervised IE is to define a terminology and to use it to manually create a small amount of labeled documents creating a *training corpus*. The training corpus is used to train an *automatic labeler* which learns to extract the information samples from yet unlabeled documents automatically. This method poses two drawbacks: The first problem which we will call the *corpus creation cost problem* is the fact that the creation of the training corpus still has to be performed by a human worker. The second problem is how to know at what point we have a sufficiently shaped training corpus to finish manual annotation and to start labeling completely automatically. One aspect of this question is if we have sufficiently annotated documents to represent the whole bandwidth of possible documents, which we call the *corpus size problem*. Another aspect is if the so far labeled documents are annotated correctly, which we call the *corpus error problem*.

One way for reducing the corpus creation costs is to use the automatic labeler already during the corpus creation process. The worker has to correct the suggested annotations and to add those omitted by the automatic system. This *guided annotation* phase has to last as long as enough documents are labeled so that the automatic labeler can process the remaining documents without manual support. Another approach to reduce costs is using untrained and hence cheaper workers. This can lead to a reduction of labeling quality as the lack of domain expert knowledge can end up in mistakes in the labeling process. Mistakes in the training corpus will result in poor quality of the trained automatic labeler and subsequently to poor overall results (*corpus error problem*). It would be beneficial if there were a way to verify the annotations in the so far labeled documents. One approach is to use a *knowledge base* provided by a domain expert which includes invariants defined according to the individual information types, e.g. some information instances should only appear once in a document, or multiple occurrences in the same document should have the same values. Further aspects described by the knowledge base are relationships among the different information types, like covariances between numerical values or confidences between qualitative information types. Using this knowledge base irregularities in the labeled documents can be detected and presented to the manual annotator to check the correctness of the elements in doubt. Another possible way to check the corpus' validity is to use the labeler on the training corpus itself and analyze the disparity in the sets of manually and automatically generated information instances in the same documents.

The corpus size problem covers the problem if the documents in the training corpus represent all the characteristics the still unlabeled documents contain. The automatic labeler will not be able to cope with information characteristics that do not appear in the labeled documents. Therefore, there has to be a sufficient amount of examples for every representation of information type in the training corpus. Usually, each new document is taken randomly from the set of unlabeled documents. It is possible using expert background knowledge and the so far trained automatic labeler to influence the selection process to choose documents from the set of unlabeled ones which will hopefully increase the diversity of information representation in the corpus. When applying the automatic labeler on a big portion of unlabeled documents the resulting annotations can be compared to the existing manual annotations and to the given background knowledge. If the automatic labels show contradictions with this knowledge or if they exhibit a different distribution from the annotations from the training corpus, this indicates a different kind of document type which should be added to the training corpus.

3 Specific Problem

Echogerät: Vivid Seven: LVDd = 44mm (< 57); LVDs = 29mm; ESVI = 22ml/m²; LVPWd = 9mm (< 11); LVEF > 55% (visuell geschätzt);LVFS = 35% (> 24); LADs = 31mm (< 40); Ao-root = 40mm (< 40); Ao-asc = 37mm (< 38); AV-Vmax = 1,4m/s (< 1,8); TR-Vmax = 2,6m/s; sPAP 33mmHg (< 35); PE_dia = 4mm; Transthorakale Echokardiographie: Schallbarkeit: Parasternal und apikal gut. Sinusrhythmus. Frequenz 99/Min. Aorta: Wurzel normal weit (40mm). Aorta aszendens normal weit (37mm). Unauffälliger morphologischer Befund an den Herzklappen. Physiologische Trikuspidalinsuffizienz. Normaler sPAP (ca. 28 - 33mmHg). Normal großer linker Ventrikel (LVDd 44mm) und linker Vorhof (LA 31mm). Kein Nachweis regionaler linksventrikulärer Wandbewegungsstörungen. Normale linksventrikuläre systolische Funktion, LVEF > 55% (visuell geschätzt). Septum 9mm, Hinterwand 9mm. Bei Tachykardie diastolische Funktion nicht sicher beurteilbar. Normal großer rechter Ventrikel. Kleiner Perikarderguss ohne hämodynamischen Auswirkungen. Echofreier Raum diastolisch 4mm. Normale antegrade Flussgeschwindigkeiten über allen Herzklappen. Normal große Herzhöhlen.

Figure 1: Example of an echocardiography report

The data we are dealing with are German echocardiography reports which are given as plain text files. An example of a typical report can be seen in figure 1.

We shape the terminology as a tree structure and add the tree's structure information as additional knowledge to each terminology entry, extending the entries to $t_i = (id_i, name_i, par\_id_i)$. Whenever a node is closer to the leaves, that represents finer grained information. The inheritances in the terminology can be exploited later by an automatic labeler. The terminology is constructed by a worker by defining to the best of his knowledge which information seems relevant and thus creates nodes in the terminology tree. After the analysis of a small amount of documents the created terminology is corrected by a medical expert. An extract of the thereby created terminology is shown in the upper part of figure 2.

The manual annotation process is performed using the TextMarker tool [Kluegl *et al.*, 2009], a tool for text annotation based on the UIMA framework [Ferrucci and Lally, 2004]. An excerpt of the annotations can be seen in the lower part of figure 2. Using this completely manually annotated corpus we train a labeler to support the annotation process by extracting preliminary entities from the documents automatically. One crucial requirement for the chosen labeler is a short training and testing time (both in the range of at maximum one minute) as it has to be used fre-

id	name	parent	norm	type	abnormal
1	Messwerte Echogerät Vivid Seven	0			
2	M-Mode- & 2D-Daten	1			
3	LVDd	2			
4	= numerisch in mm	3	x < 57.0	numeric	
5	ESVI	2			
6	= numerisch in ml/m²	5	21.0 < x < 33.0	numeric	
7	Doppler-Daten	1			
8	AV-Vmax	7			
9	= numerisch in m/s	8	0.9 < x < 1.8	numeric	
10	AV-Pmax	7			
11	= numerisch in mmHg	10	x < 30.0	numeric	
12	Transthorakale Echokardiographie	0			
13	Schallbarkeit	12			
14	apikal	13		single choice	
15	gut	14			
16	schlecht	14			x
17	parasternal	13		single choice	
18	gut	17			
19	schlecht	17			x
20	Aorta	12			
21	Wurzel	20		single choice	
22	normal	21			
23	erweitert	21			x
24	Klappeninsuffizienzen	12		single choice	
25	ja	24			x
26	nein	24			

id	covered text	offset
1	Echogerät: Vivid Seven	195 – 217
3	LVDd	219 – 223
4	= 44mm (< 57)	225 – 237
5	ESVI	252 – 256
6	= 22ml/m²	258 – 267
12	Transthorakale Echokardiographie	488 – 520
13	Schallbarkeit	535 – 548
14	apikal	567 – 573
17	parasternal	551 – 562
15	gut	575 – 591
18	gut	575 – 591
…	…	…

Figure 2: up.: terminology excerpt ; lo.: annotation list excerpt

quently to support the manual annotation process. We developed a labeler based on entity extraction by keyword matching and assignment of correct terminology IDs by window based disambiguation. A pseudo-code description is presented in algorithm 1.

Algorithm 1 label(newDoc)

terminology T = {t_1, .., t_n}; term t_i = (id_i, par_id_i)
training corpus D = {d_1, .., d_n}; docu. d_i = ($text_i$, A_i)
annotation a_{ij} = (t_{ij}, $text_{ij}$, $offsets_{ij}$)
annotations A_i = { a_{i1}, .., a_{in} }; A = $\cup_i A_i$
keywords = {}; entities = {}; wSize = 1

for all (d_i in D) **do**
  for all (a_{ij} in $d_i.A_i$) **do**
    keywords.add(a_{ij}.text.norm());
for all (k_i in keywords) **do**
  entities.addAll(newDoc.text.norm().findAll(k_i))
unassEnt = entities.copy()
for all (e_i in unassEnt) **do**
  possT$_i$ = { t | ∃ a ∈ A : t = a.t ∧ a.text.norm() = e_i.text }
  if (|possT$_i$| = 1) **then**
    e_i.t = possT$_i$.getOnlyElement()
    unassEnt.remove(e_i)
sent = newDoc.text.splitIntoSentences()
repeat
  for all (e_i in unassEnt) **do**
    sentWin = ∩ all sent.bounds in wSize around e_i
    possT$_i$ = { t | ∃ a ∈ A : t = a.t ∧ a.text.norm() = e_i.text }
    surrEnt = { e | e ∈ entities ∧ e.bounds ⊂ sentW }
    surrEnt.orderByTextDistTo(e_i)
    for all (e_j in surrEnt) **do**
      if (e_j ∈ unassEnt) **then**
        possT$_j$ = { t | ∃ a ∈ A : t = a.t ∧ a.text.norm() = e_j.text }
      else
        possT$_j$ = { e_j.t }
      parentT = { t_i ∈ possT$_i$ | ∃ t_j ∈ possT$_j$: t_i = t_j.par_id }
      if (|parentT| = 1) **then**
        e_i.t = parentT.getOnlyElement()
        unassEnt.remove(e_i)
        wSize = 0
        continue with next e_i
      childT = { t_i ∈ possT$_i$ | ∃ t_j ∈ possT$_j$: t_i.par_id = t_j }
      if (|childT| = 1) **then**
        e_i.t = childT.getOnlyElement()
        unassEnt.remove(e_i)
        wSize = 0
        continue with next e_i
  wSize = wSize + 1
until (wSize=wmax) ∨ unassEnt.isEmpty()

Algorithm Overview

The overall idea of the algorithm is the identification of entities (i.e. sometimes single tokens, sometimes aggregations of multiple tokens) of a text as information containing fragments. A terminology entry has to be assigned to each of the extracted entities. This is done by comparing the covered text of the entities with the annotations from the training corpus. If an unassigned entity has the same covered text as an annotation from the training corpus, is this a reference for assigning this entity to the same terminology entry. First all entities with only one possible meaning are assigned to their corresponding entries, then in an iterative way, all remaining entities get assigned, exploiting the information of other already assigned entities in the textual surroundings. If the intersection of the possible terminology entries for an unassigned entity with the possible terminology entries for a nearby entity results in only one possible element, is this entry assigned to the inspected entity. The window defining an entity's surrounding can be expanded so that in the end all entities are assigned or no further assignments take place.

Algorithm Details

The first step is to take the covered text of all phrases from the whole training corpus constructing a *keyword* list $K = \{k_1, .., k_n\}$ with keywords k_i. All occurrences of numbers in the keywords are replaced by wild-cards in a normalization step. In the next step we construct a list of all occurrences in the processed document (also normalized) where any elements of the keyword list appear. All entities in this list are declared as *unassigned entities*, i.e. annotations for which the corresponding terminology entry still has to be identified. Now each document text is split into sentences at predefined sentence boundaries like periods or semicolons. Then, all unassigned entities for which exactly one terminology entry exist, which has at least one annotation that has the same covered text as the unassigned entity, gets this terminology entry assigned to and gets removed from the list of unassigned entities. Now an iterative process begins. It starts with a window of the size of the sentence around the respective unassigned entity and gets expanded by the adjacent sentences, if no new entities get assigned in the current iteration round. In each iteration all unassigned entities are compared with all other entities in the window around them. The entities are ordered with respect to their textual distance to the unassigned entity. If, for an ambiguous entity e_i with possible assignments t_i = $\{t_{i,1}, .., t_{i,n}\} \subset T$ and another entity e_j with possible assignments t_j = $\{t_{j,1}, .., t_{j,n}\} \subset T$, there exists exactly one $t_{i,x}$ for arbitrary (not necessarily all) $t_{j,*}$ which is a parent of any of the possible entries of the other entry so that $t_{i,x}$ = $t_{j,*}$.par_id (*parent relationship*), or which is a child of any of the possible entries of the other entry so that $t_{i,x}$.par_id = $t_{j,*}$ (*child relationship*), then e_i gets assigned $t_{i,x}$. If an entry is assigned, testing for the current entity is aborted and it is removed from the list of unassigned entities. As with each iteration there will be more assigned entities which provide information for other still unassigned entities, the process is repeated until there are no further assignments generated. As an entity could be related to an entity mentioned in a former or a successive sentence the process is repeated with a larger window until again no further entities get assigned. When the maximum window size has been reached (in our case 5) the extraction process is terminated.

Error Correction

The trained labeler cannot only be used to preannotate unlabeled documents, it can also be used to check the validity of the corpus. The labeler, which was trained with the training corpus can be utilized to automatically annotate the training corpus itself. All differences in the manual and the automatic annotations are due to contradictions in the training corpus thus indicating manual annotation errors. Another way for error checking is the use of background knowledge. We add the information to each entry if the respective information is of quantitative (numeric) or qualitative nature. If qualitative, additional information is added whether the different possible types of information the entry can represent are mutually exclusive or if the node represents a multiple choice value. Additionally for the children of single choice nodes we are given the information if a child node represents an abnormal state for its parent. For quantitative information the expert adds a norm value interval. With this additional information the terminology entries are extended to $t_i = (id_i, name_i, par\_id_i, type_i, norm\_int_i, abnorm_i)$. Furthermore, a medical expert provides a set of entry pairs $R = \{r_1, ..., r_n\}, r_i = (t_i^1, t_i^2)$ which denote medical confidence rules, i.e. the existence of an abnormal measurement for information t_i^1 implies that there should also be an abnormal measurement for information t_i^2. Using this knowledge we can check in all annotated documents if the annotations satisfy the given confidence rules. The rules can not only be applied onto the training corpus but also onto automatically annotated documents. Disagreements between automatic annotations and the confidence rules can be an indication of errors in the training corpus.

4 Case Study

We instructed a student to serve as the worker in our manual annotation process. During the corpus creation phase, in which the terminology was created simultaneously, the annotation of one document took about 45 minutes. This time was reduced in the course of further labeling to about 30 minutes, as the terminology had to be expanded less and the worker gained more experience with the annotation tool. We decided that our trained labeler could be reasonably used with a training corpus size of about 15 completely manually annotated documents. Because of the short training and testing times our described labeler could be comfortably used by the worker as automatic support before annotating each document for creating provisional annotations. During the guided annotation phase it took the worker only about 8 minutes for the annotation of one document. In the error correction phase we compared the manual annotations with automatically generated ones, and performed additional check-ups using the background knowledge about correlations between the different information types. We did these error checks not after every labeled document but only in two correction runs. We corrected after 40 and after 80 labeled documents. By making the corrections during the second correction run the f-value of our labeler on the training corpus itself was raised in a two fold test on the annotated documents from 0.96 up to 0.992. After 80 annotated documents we ran the automatic labeler on 2500 unlabeled documents. The annotations from this automatic corpus were analyzed with our set of correlation rules. All documents which heavily contradicted were taken into a set of documents which had to be reworked manually. It turned out that in those documents the texts had a different structure with different keywords than in the previous corpus which lead to false positives thereupon leading to the contradictions to our knowledge base.

5 Conclusion

We presented an approach for creating an annotated training corpus for the training of an automatic labeler for unstructured medical texts. We described a labeler which could be trained and tested very quickly to support the annotations process. The proposed approach was applied onto a set of echocardiography records. It is problematic to verify the quality of the processed set as there is no satisfactory way to check the correctness of automatically annotated documents other than manual checking or using sparse background knowledge. As medical records are rarely published, it is also difficult to compare our results to those from other groups. We can state that the proposed approach worked well in our domain and may work in further domains we want to work on. Due to verification difficulties we plan to expand the possibilities to counter check with further background knowledge. In further projects we want to turn to more complex domains, like sonography and endoscopy records, which contain less structured sections. Depending on the increased complexity, we plan to enrich our labeler with more profound text analysis techniques like POS tagging and shallow parsing.

This work was supported by grants from the Bundesministerium für Bildung und Forschung (BMBF01 EO1004).

References

[Aronson, 2001] A.R. Aronson. Effective mapping of biomedical text to the umls metathesaurus: The metamap program, 2001.

[Chung and Murphy, 2005] J. Chung and S.N. Murphy. Concept-value pair extraction from semi-structured clinical narrative: a case study using echocardiogram reports. *AMIA Annu Symp Proc*, 2005.

[Denny and Peterson, 2007] J. C. Denny and J. F. Peterson. Identifying qt prolongation from ecg impressions using natural language processing and negation detection. In *MedInfo*, pages 1283–1288, 2007.

[Ferrucci and Lally, 2004] D. Ferrucci and A.D.A.M. Lally. Uima: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.

[Friedman, 2000] C. Friedman. A broad-coverage natural language processing system. In *AMIA*, pages 270–274, 2000.

[Hripcsak *et al.*, 1995] G. Hripcsak, C. Friedman, P. Alderson, W. Dumouchel, S. Johnson, and P. Clayton. Unlocking clinical data from narrative reports: a study of natural language processing. 1995.

[Kluegl *et al.*, 2009] P. Kluegl, M. Atzmueller, and F. Puppe. Textmarker: A tool for rule-based information extraction. In *Proc. UIMA, 2009 Conference of the GSCL*, 2009.

[Mamlin *et al.*, 2003] Burke W Mamlin, Daniel T Heinze, and Clement J McDonald. Automated extraction and normalization of findings from cancer-related free-text radiology reports. *AMIA Annu Symp Proc*, 2003.

[Meystre *et al.*, 2008] S.M. Meystre, G.K. Savova, K.C. Kipper-Schuler, and J.F. Hurdle. Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research. *IMIA Yearbook 2008: Access to Health Information*, 2008:128–144, 2008.

[Savova *et al.*, 2010] G. Savova, J. Masanz, P. Ogren, J. Zheng, S. Sohn, K. Kipper-Schuler, and C. Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *JAMIA*, 17(5):507–513, 2010.