Applying In-Memory Technology for Automatic Template Filling in the Clinical Domain

Working Notes on Task 2 of the ShARe/CLEF eHealth Challenge 2014

Konrad Herbst[2], Cindy Fähnrich[1],
Mariana Neves[1], and Matthieu-P. Schapranow[1]

[1] Hasso Plattner Institute
Enterprise Platform and Integration Concepts Chair
August-Bebel-Str. 88,
14482 Potsdam, Germany
{cindy.faehnrich|mariana.neves|schapranow}@hpi.de
[2] University of Heidelberg
Institute of Pharmacy and Molecular Biotechnology
Im Neuenheimer Feld 364,
69120 Heidelberg, Germany
k.herbst@stud.uni-heidelberg.de

Abstract. We present a research prototype for systematic template filling based on in-memory database technology. Entity extraction and normalization is based on domain-specific dictionaries and customized rules set building on top of related work of the medical field. The prototype called *HPI* proves feasibility of in-memory technology to enhance workflows in the field of efficient text processing and analysis. With our approach, the iterative process of dictionary and rule refinement for enhancing text analysis results shifts from a time-consuming task with long waiting hours to a continuous workflow. In the context of the challenge's task, our prototype achieves an overall average accuracy of 0.769 and an overall F_1 measure of up to 0.323.

Keywords: Medical Reports, Template Filling, In-Memory Technology, Entity Recognition, Text Extraction

1 Introduction

Professional health care requires a constant documentation of all patient-related data, such as history of clinical events. This clinical data is stored in a human-readable format, such as text files, since it supports the daily work of the clinical personnel. This data is only available in an unstructured format, which makes its automatic processing a complex task. However, for the sake of fault prevention, comparison, performance optimization, and subsequent clinical research, the important information must be efficiently extracted from the unstructured

data for further processing. This task requires methods from Information Extraction (IE), which is a specific subdomain of Natural Language Processing (NLP).

The second task of the 2014 CLEF eHealth challenge requires the extraction of information from unstructured clinical data to fill specific templates, i.e. fixed sets of different semantic classes depending on the IE purpose [2, 3, 6]. The following classes are required to be identified: Negation Indicator (NI), Subject Class (SC), Uncertainty Indicator (UI), Course Class (CC), Severity Class (SV), Conditional Class (CO), Generic Class (GC), Body Location (BL), Doctime Class (DT), and Temporal Expression (TE). Within these classes, values can be stored either as recognized text span, i.e. where the entity was determined within the input text, or as inferred concept normalization. A lexical cue value describing the found occurrence of the entity within the input text can be determined for all class types except for DT.

We as team HPI participated in the context of a student internship in this challenge. We designed a research system incorporating lasted In-Memory Database (IMDB) technology to enable systematic filling of templates of the required classes using unstructured data from Electronic Medical Records (EMR). IMDB technology has proven to have major advances for analyzing big enterprise and medical data, e.g. to support medical doctors in identifying better treatments for cancer patients and other fields of life sciences [13, 8, 14]. Thus, IMDB supports a) the interactive processing of EMR data, which b) enables fast, iterative design of productive systems for TE and its analysis. We rely on a columnar IMDB and make use of the built-in Text Analysis (TA) functionality for our research prototype. Additionally, we complement data provided for training with additional external data sources and extract relevant entities as described in Sect. 2.1.

2 Methods

In the following, we describe data used in our system, its architectural details, and highlight the advantages of using IMDB technology.

2.1 Data

For the training phase, we used a data set of 300 documents taken from version 2.5 of the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC II) database [5, 9]. This data is comprised of a corpus and annotations of de-identified clinical reports from intensive care patients from the United States of America (USA). These reports are classified into four types: discharge summary, echo report, electrocardiogram report, and radiology report. All documents are unstructured text documents, i.e. they are written in natural language without specific formatting.

In addition to the training data, we integrated the SNOMED Clinical Terms (SNOMED CT) data from the Unified Medical Language System (UMLS) ver-

sion 2013AB to improve our entity recognition capabilities for mentions of diseases and body locations [15]. From this database, we used all concepts with a semantic type that is related to a disease, disorder, or body location. Tab. 2.1 provides a detailed overview of what concepts and semantic types we have incorporated. These concepts sum up to a data set of >183k concepts, i.e. entities that can be used for entity recognition in the training data summing up to more entries than the complete SNOMED CT data set.

Table 1: Overview of the SNOMED CT subset incorporated in our research prototype. We selected concepts assigned to semantic types that are either related to Diseases/Disorders (DD) or Body Locations (BL). The overall amount of concepts used from SNOMED CT sums up to >183k concepts that are used for entity recognition in the training data.

Semantic Type (Slot Type)	# Concepts
Disease or Syndrome (DD)	34,600
Injury or Poisoning (DD)	26,703
Neoplastic Process (DD)	9,082
Congenital Abnormality (DD)	6,337
Pathologic Function (DD)	5,364
Mental or Behavioral Dysfunction (DD)	2,745
Signs and Symptoms (DD)	2,734
Acquired Abnormality (DD)	1,795
Anatomical Abnormality (DD)	1,475
Cell or Molecular Dysfunction (DD)	382
Experimental Model of Disease (DD)	3
Body Part, Organ, or Organ Component (BL)	59,027
Body Location or Region (BL)	10,797
Body Space or Junction (BL)	6,994
Tissue (BL)	4,130
Body Substance (BL)	2,793
Cell (BL)	2,602
Cell Component (BL)	2,602
Embryonic Structure (BL)	2,110
Body System (BL)	787
Anatomical Structure (BL)	111
Fully Formed Anatomical Structure (BL)	8
Total	183,181

2.2 Using In-Memory Database Technology for Data Processing

For accomplishing the challenge's task, we designed a research prototype incorporating the latest IMDB technology. It enables us to store and process structured

and unstructured data within a single system as it has several building blocks as presented by Plattner [7]. In the following paragraphs, we introduce selected building blocks and how we benefit from them for accomplishing our task.

Relevant Data Kept in Main Memory IMDB technology enables fast access of required data directly from main memory. This contrasts to most traditional approaches processing data from files that reside on disk space and must be loaded into main memory. When thinking of the ever-increasing amounts of data, this strategy will not be feasible anymore in the long run. Therefore, IMDB technology offers us an alternative processing strategy that addresses performance requirements of our application.

Lightweight Compression Those techniques refer to a data storage representation that consumes less space than its original pendant. A columnar database storage layout supports such lightweight compression techniques, e.g. dictionary encoding which maps all unique values to a uniform format [7]. For example, suppose we have a list of people as data set where one column contains the gender. For this column, there exist only two unique values, i.e. "male" and "female". With dictionary encoding, these two values are mapped to integer representations, e.g. "male"=1 and "female"=2, and stored in the column instead of the original values. This requires less storage space and also reduces the amount of data that has to be transferred from and to main memory.

Multi-Core and Parallelization Modern system architectures are designed to provide multiple CPUs with each of them having separate cores. This capacity should be fully exploited by parallelizing application execution to achieve maximum processing speed. The incorporated IMDB platform supports this and provides built-in parallelization. With that, we do not need to apply parallelization strategies on our own but still have maximum runtime performance in processing our input data.

Entity and Feature Extraction Any kinds of text, such as the medical reports that have to be processed in this challenge, are considered as unstructured data. Thus, it cannot be processed automatically unless a machine-readable data model exists for automatic interpretation, e.g. a semantic ontology. Our incorporated IMDB platform offers a range of features for text processing, of which the relevant ones for us are those for entity and feature extraction. Entity and feature extraction refers to the identification of relevant keywords and names of entities from documents. Dictionaries and individual extraction rules can customize this. Dictionaries list one or more entity types, each of which containing any number of entities that in turn contain a standard form name and any number of synonyms. Extraction rules use formal syntax to define entities of a specific type. This allows formulating patterns that match tokens by using a literal string, a regular expression, a word stem, or a word's part of speech.

2.3 System Design

Fig. 1 presents our system architecture in Functional Modeling Concepts (FMC) notation [4]. Medical reports as test data and a dictionary that has been generated from the training data in advance serve as input for our system. This data is imported once into our IMDB. The input template documents must now be automatically filled with concrete values for cue and normalization attributes. The system itself is divided into two components: Our IMDB platform, which performs among others linguistic pre-processing tasks, e.g. entity extraction via dictionaries, and a Python module for template filling.

In-Memory Database Relevant data is imported into our IMDB. The data is comprised of the medical reports whose templates must be filled, the SNOMED CT subset, and a list for each slot type with entities that have been extracted from the training data before. From this data, we create scientific medical dictionaries and add individual extraction rules to facilitate entity recognition and extraction of the different slot types.

Dictionaries We build customized dictionaries to identify slot types NI, SC, UI, CC, SV, CO, GC, BL, and TE in the given medical reports. For extraction and normalization of DD and BL slot types, we compile a dictionary based on the imported SNOMED CT data set. Entities of remaining slot types are extracted and normalized by a dictionary derived from training data. Fig. 2a depicts such a dictionary in XML format. Entities can easily be organized into categories, normalized by a standard form and enriched by additional variant definitions. The given example lists the semantic type *Body Part, Organ, or Organ Component* in blue letters. Afterwards, the concept definition with its normalization, i.e., the standard form, is defined in black letters. Finally, possible entities owning the defined normalization are listed in yellow letters. As a result, the phrase *skeletal muscle structure of abdomen* has the normalization *C0000739* and will be assigned to the BL slot type when detected in any text document.

CGUL Rules We define extraction rules in Custom Grouper User Language (CGUL) to identify DT slot types [11]. CGUL is a sentence-based language that allows pattern matching by using character or token-based regular expressions combined with linguistic attributes to define custom entity types. Fig. 2b shows two example CGUL rules for extracting entities that have *before* and *before_overlap* as normalization. By using Part-of-Speech (POS) tags in the rules, we can access and extract the grammatical tense of a sentence. In the given examples highlighted in purple color, we want to identify structures that first contain a noun (*Nn*) after which comes a verb in either past (*V-Past*) or past participle (*V-PaPart*) tense. Means to identify nouns, verbs, and tenses are provided by default by our IMDB platform.

Entity Recognition and Extraction With the created dictionaries and CGUL rules at hand, we can trigger the actual process of entity extraction within our IMDB.

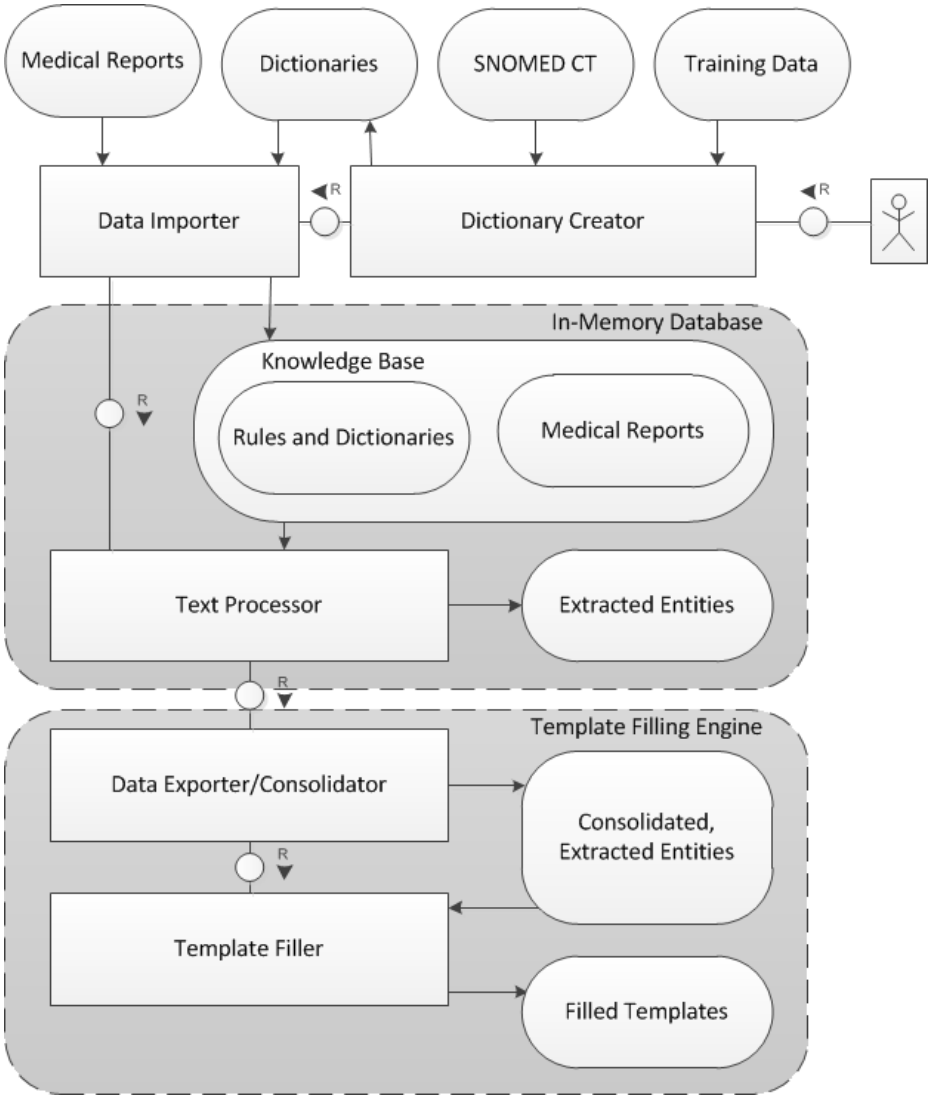

Fig. 1: Architecture of our research prototype in Fundamental Modeling Concepts (FMC) notation. In addition to small preprocessing steps of dictionary creation and data import, our prototype consists of the two main components In-Memory Database (IMDB) and a Template Filling Engine. Main processing is conducted inside these two components.

(a)

(b)

Fig. 2: Examples for (a) dictionary entry and (b) CGUL rules for entity recognition. An entry in a dictionary is comprised of the slot type, its normalization format and concrete entities listed. CGUL rules allow entity recognition via matching lexicographical patterns, e.g. by identifying nouns (Nn), verbs (V), or tenses.

For that, we create a full text index on the medical reports for which we have to fill out the templates [10]. The full text index is automatically managed by our IMDB, which performs linguistic processing, i.e., language and encoding identification, segmentation, case normalization, stemming, and tagging, and entity and fact extraction based on the provided dictionaries and CGUL rules [12]. The result of this process is a dedicated database table that contains the extracted entities that have been found in the medical reports, their normalization, slot type, and location within the document. These details can be directly used for template filling.

Template Filling Our template filling engine is based on Python v. 2.7.7 and takes extracted entities together with their normalization, slot type, medical report it occurred in, and location within the medical report, i.e. text spans, as input. These details are associated to the corresponding templates, which requires matching the text spans identified by our approach. The DD mentions provided by default in the test data are used as "anchor" to determine entities of the same template. If this has been accomplished, the templates are filled with the corresponding cues and normalizations.

3 Conducted Experiments

In the following, we present experiments conducted in terms of data and evaluation metrics used. We provide experiment results according to the presented metrics and discuss relevant findings.

3.1 Data and Metrics Used

For evaluating the performance of our system, we used a test data set provided by the challenge. Analogously to the initial training data, this data set is comprised of a set of 133 medical reports with template documents assigned. In contrast to the training data, the templates' attributes, i.e. cue and normalization values for each slot type, are empty. In our experiments, we aim at filling both cue and normalization values and by that to participate in tasks 2a and b of the challenge.

We use accuracy and F_1 measure as common measures used in pattern recognition and information retrieval for evaluation of the derived normalization and cue values, respectively [1]. We determine performance for the overall result set and per slot type. Eq. 1 defines the computation of accuracy for a given set of normalization values N as fraction of the amount of slot values for which a correct normalization has been derived and the overall amount of slot values for which a normalization has been derived.

$$Accuracy(N) = \frac{|N_{correct}|}{|N|} \tag{1}$$

Eq. 2, Eq. 3, and Eq. 4 depict computation of F1 measure to assess quality of the detected cue values, which is the harmonic mean of precision and recall. With regards to examining performance for a concrete slot type or their overall set, C is the set of all cue values detected by our approach, whereas C_{true} contains all true cue values. $C_{correct}$ is the set of all cue values that have been correctly identified by our approach and is also expressed as $C_{correct} = C_{true} \cap C$. The definition of the term "correct" varies for strict and relaxed evaluation. The former checks if a derived cue value equals the correct one, whereas the latter still considers a cue value as correct if it overlaps with the true value. Precision depicts the fraction of retrieved instances that are relevant, i.e., in this context how many of the true cue values have been identified by our approach. Recall depicts the fraction of relevant instances that are retrieved, i.e. how many of the cue values identified by our approach are contained in the set of "true" cue values.

$$F_1(C) = \frac{2 \times Recall(C) \times Precision(C)}{Recall(C) + Precision(C)} \tag{2}$$

$$Recall(C) = \frac{|C_{correct}|}{|C_{correct}| + |C_{true} \setminus C|} \tag{3}$$

$$Precision(C) = \frac{|C_{correct}|}{|C_{correct}| + |C \setminus C_{true}|} \tag{4}$$

3.2 Results and Discussion

Table 2: Summarized results for task 2a with results for accuracy, F_1 measure, precision, and recall.

Normalized Slot Type	Accuracy	F_1 Measure	Precision	Recall
Norm_BL	0.494	0.072	0.121	0.051
Norm_CC	0.899	0.250	0.174	0.445
Norm_CO	0.819	0.317	0.209	0.658
Norm_DT	0.060	0.060	0.060	0.060
Norm_GC	1.000	0.000	0.000	0.000
Norm_NI	0.762	0.265	0.370	0.207
Norm_SL	0.976	0.356	0.294	0.450
Norm_SV	0.914	0.310	0.273	0.359
Norm_TE	0.864	0.000	0.000	0.000
Norm_UI	0.906	0.410	0.327	0.549
Overall	0.769	0.128	0.136	0.121

Table 3: Summarized results for task 2b with results for F_1 measure, precision, and recall for both strict and relaxed evaluation.

Cue Slot Type	F_1 Measure		Precision		Recall	
	strict	relaxed	strict	relaxed	strict	relaxed
Cue_BL	0.098	0.363	0.165	0.611	0.070	0.258
Cue_CC	0.210	0.283	0.145	0.196	0.378	0.510
Cue_CO	0.076	0.317	0.050	0.209	0.157	0.658
Cue_GC	0.096	0.139	0.056	0.081	0.325	0.470
Cue_NI	0.332	0.465	0.349	0.488	0.317	0.444
Cue_SC	0.100	0.151	0.057	0.086	0.411	0.620
Cue_SV	0.345	0.396	0.293	0.336	0.420	0.483
Cue_TE	0.000	0.000	0.000	0.000	0.000	0.000
Cue_UI	0.138	0.306	0.094	0.209	0.258	0.572
Overall	0.159	0.323	0.154	0.314	0.163	0.332

The results achieved by our prototype are summarized in Tab. 3.2 and Tab. 3.2 for tasks 2a and 2b, respectively. For many of the slots, our results for task 2b, i.e. cue values derived, were quite lower than the ones obtained for task 2a, i.e. normalized values derived. For instance, we achieved 90-100 percent of accuracy for the slot types CC and GC, but only 21 and 14 percent F_1-measure, respectively, for the relaxed evaluation of task 2b. Although this is expected, as exact (or relaxed) mention spans are harder to be correctly extracted than the corresponding normalized values, we still investigate possible mistakes on the offsets in our submissions and future error analysis will shed some light on the discrepancies between the results for both tasks.

Our strategy for the BL slot, which had relied on the dictionaries derived from the SNOMED CT terminology, achieved 50 percent of accuracy. A future error analysis will also show whether false negatives were due to concepts that are not present in the SNOMED CT terminology, to missing synonyms for existing concepts or on the matching approach that was used. Nevertheless, the relaxed evaluation of task 2b shows that our dictionary matching approach provides good precision, i.e. 60 percent, given the complexity of the anatomical nomenclature.

Extraction of values for slot type DT was a hard task and results were quite low for all teams. This is because it requires a more careful analysis of the language, such as analyzing verb tenses and time expressions. However, we believe that our approach of using CGUL rules is appropriate for extracting this information but more rules should be created for this purposes as well as a revision of the existing ones.

Therefore, the used dictionaries and rules instead of the underlying IMDB system induce the presented results. If those dictionaries are refined, e.g. by including other data sources than SNOMED CT or adapting extraction rules, we are convinced that the overall performance of our system will improve.

However, the focus of this work is rather on showing the general applicability and feasibility of in-memory technology for processes that involve processing and analysis of unstructured text. One iteration to improve text analysis results, starting with refining dictionaries and ending with receiving the final results, i.e. the filled templates from the test data, takes minutes with our system instead of hours or days with traditional approaches. This proves that in-memory technology provides advantages also for the field of information extraction and can contribute to establishing efficient and alternative processing strategies in that area.

4 Conclusion

The ShARe/CLEF eHealth challenge 2014 aims to facilitate the research on information extraction within the biomedical domain. As follow-up to 2013's challenge, participants were asked to identify semantically related mentions to disorder mentions and fill out templates with normalization and cue values for the detected entities.

In the context of a student internship, we designed a research prototype for entity extraction based on IMDB technology that proves feasibility for efficient text processing. Evaluation results show that our rules and dictionaries currently applied require optimization by refining dictionaries or extraction rules. However, our prototype allows us to extend existing extraction rules and dictionaries in a constant manner and to verify them instantly. Thus, the task of iterative improvement of text analysis results becomes a continuous process.

Bibliography

1. Christen, P., Goiser, K.: Quality and Complexity Measures for Data Linkage and Deduplication. In: Quality Measures in Data Mining, pp. 127–151. Springer (2007)
2. Elhadad, N., et al.: The ShARe Schema for the Syntactic and Semantic Annotation of Clinical Texts
3. Kelly, L., et al.: ShARe/CLEF eHealth Evaluation Lab 2014 (2014), Springer-Verlag
4. Knöpfel, A., Grone, B., Tabeling, P.: Fundamental Modeling Concepts: Effective Communication of IT Systems. John Wiley & Sons (2006)
5. Mowery, D.L., Velupillai, S.: Task 2 Data Set of the ShARe/CLEF eHealth Challenge 2014. `http://clefehealth2014.dcu.ie/task-2/2014-dataset` [retrieved: Jun, 2014] (Jun 2014)
6. Mowery, D.L., Velupillai, S.: Task 2: ShARe/CLEF eHealth Evaluation Lab 2014. `http://clefehealth2014.dcu.ie/task-2` [retrieved: Jun, 2014] (Jun 2014)
7. Plattner, H.: A Course in In-Memory Data Management: The Inner Mechanics of In-Memory Databases. Springer, 1st edn. (2013)
8. Plattner, H., Schapranow, M.P. (eds.): High-Performance In-Memory Genome Data Analysis: How In-Memory Database Technology Accelerates Personalized Medicine. Springer-Verlag (2014)
9. Saeed, M., et al.: Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): A Public-Access ICU Database. Clinical Care Medicine 39, 952–960 (2011)
10. SAP AG: SAP HANA SQL and System Views Reference. `http://help.sap.de/hana/SAP_HANA_SQL_and_System_Views_Reference_en.pdf` [retrieved: Jun, 2014] (Jun 2011)
11. SAP AG: Text Data Processing Language Reference Guide. `https://help.sap.com/businessobject/product_guides/boexir4/en/sbo401_ds_tdp_lang_ref_en.pdf` [retrieved: Jun, 2014] (Jun 2011)
12. SAP AG: SAP HANA Text Analysis Language Reference Guide V. 1.0. `http://help.sap.com/hana/SAP_HANA_Text_Analysis_Language_Reference_Guide_en.pdf` [retrieved: Jun, 2014] (May 2014)
13. Schapranow, M.P., et al.: Mobile Real-time Analysis of Patient Data for Advanced Decision Support in Personalized Medicine. In: Proceedings of the 5th Int'l Conf on eHealth, Telemed, and Social Medicine (2013)
14. Schapranow, M.P., Häger, F., Fähnrich, C., Ziegler, E., Plattner, H.: In-Memory Computing Enabling Real-time Genome Data Analysis. Advances in Life Sciences 6(1–2) (2014)
15. U.S. National Library of Medicine: Unified Medical Language System (UMLS). `http://www.nlm.nih.gov/research/umls/` [retrieved: Jun, 2014] (Jul 2013)