TeamUEvora at CLEF eHealth 2014 Task2a

João Sequeira, Nuno Miranda, Teresa Gonçalves, Paulo Quaresma

Computer Science Department, School of Science and Technology
University of Évora, Évora, Portugal
{jsequeira,nmiranda,tcg,pq}@uevora.pt

Abstract. We present our first participation in a ShARe/CLEF eHealth Lab contributing for task 2a. Task 2 is an extension of the 2013 lab task 1 and consists of information extraction from clinical texts for Disease/Disorder Template Filling; task 2a aims at predicting each attribute's normalization value.
This work constitutes a preliminary approach to the problem of extracting and handling information from clinical texts. More than getting a good result, our priority was to get a first hint on the questions and problems that are posed within this area.
For that, we developed a system that combines information from cTAKES output and the training corpus. The performance was measured using accuracy. Our system ranked 7th with an accuracy of 0.802, a F_1 of 0.214, a precision of 0.217 and a recall value of 0.211.

Keywords: Clinical texts, Template filling, Text normalization, cTAKES, Medical Informatics

1 Introduction

The ShARe/CLEF eHealth Lab 2014[1] [1,2] task 2 is an extension of the task 1 of the same lab from 2013 [3] and consists of information extraction from clinical texts with the goal of disease/disorder template filling. For each disease/disorder present in each clinical report there is a template with ten different attributes and participants have to predict the value for each attribute. There are two subtasks: 2a) assign normalization values to the ten attributes; 2b) assign cue values to the nine attributes with cues.

This is our first participation in a ShARe/CLEF eHealth Lab and we contributed to subtask 2a, building a system that uses previous implemented technologies. Being this the first time we work with medical information, our main priority is to understand the problems associated with the extraction of information in the area. In this paper we present the system architecture and the decisions made; we also present and analyse the experimental results on the training and test corpora.

[1] https://sites.google.com/a/dcu.ie/clefehealth2014/

The paper has the following structure: Section 2 introduces the task, the training and test corpora in detail and Section 3 presents the implemented system. The results are discussed in Section 4 and conclusions and a glimpse of future work are presented in Section 5.

2 Task

As said in Section 1, task 2 is an extension of the 2013 task 1 lab aiming at filling templates with attributes values and cues.

Files with empty templates for each disease/disorder (mentioned in the corresponding clinical text) were provided to the participants. Each template indicates the Unified Medical Language System Concept Unique Identifier (CUI), mention boundaries and the ten attributes needed to be filled. Each attribute has two slot types: the normalized value and the lexical cue from the sentence where the normalized value occurred. Task 2a evaluates the systems' ability to predict the normalized value for each attribute and task 2b the ability to find the right cue slot value for each attribute.

Since we participated only on task 2a (that was mandatory), our templates have default values in all the cue slots. Table 1 presents template information: a header with the file name, the cue slot of the disease/disorder and its CUI, the nine modifiers associated with the disease/disorder with normalized values (task 2a) and cue slots (task 2b) plus the `DocTime` modifier that only has a normalized value.

2.1 Description of the training and test corpora

The train and test corpora provided are composed of clinical texts from four different types: discharge summary, ECG report, ECHO report and radiology report. Their distribution in each corpus is presented in Table 2.

Analysing both corpora we can observe some differences. In the training corpus the Discharge summary type has 45.82% of documents while the remaining classes have an equal number, 18.06%; in the test corpus there are only Discharge summary documents.

3 System Architecture

This section presents the implementation of our system and the approaches taken to tackle the modifiers.

3.1 cTAKES

As said before, our system uses previous implemented technologies for clinical texts analysis and information extraction (this method was also used in task 1 [6,7,8,9,10,11,12,13,14] of 2013 ShARe/CLEF eHealth Lab).

Table 1. Template representation with the default values identified by (*).

Header		
File name Cue slot Concept Unique Identifier (CUI)		
Modifiers		
Attribute	**2a) Normalized values**	**2b) Cue slot**
Negation indicator (NI)	yes/no*	if value is yes
Subject class (SC)	patient*, family_member, other, null, donor_family_member, donor_other	if different of patient
Uncertainty indicator (UI)	yes/no*	if value is yes
Course class (CC)	unmarked*, changed, increased, decreased, improved, worsened, resolved	if different of unmarked
Severity class (SV)	unmarked*, severe, slight, moderate	if different of unmarked
Conditional class (CO)	true/false*	if value is true
Generic class (GC)	true/false*	if value is true
Body location (BL)	NULL*, CUI, CUI-less	if different of NULL
DocTime class (DT)	unknown*, before, after, overlap, before-overlap	no slot
Temporal Expression (TE)	none*, date, time, duration, set	if different of none

Table 2. Number and percentage of documents of each type in the train corpus and test corpus.

Type	**Train**		**Test**	
	no. docs	%	**no. docs**	%
Discharge summary	137	45.82	133	100.00
ECG report	54	18.06	0	0.00
ECHO report	54	18.06	0	0.00
Radiology report	54	18.06	0	0.00
Total	299	100.00	133	100.00

We used the output of the clinical Text Analysis and Knowledge Extraction System (cTAKES) [4] (version 3.1.1). cTAKES[2] is a open source linguistic tool kit from the Apache Software Foundation. Some operations done by cTAKES include:

- boundary detection;
- tokenization;
- morphological normalization;
- POS tagging;
- shallow parsing;
- negation detection;
- named entities detection with mapping to UMLS terms;
- relations detection

3.2 Modifiers

Negation and Uncertainty Indicators, Subject and Conditional Classes and Body Location. For the modifiers `NI`, `SC`, `UI`, `CO` and `BL` we extracted the information from the cTAKES output. Among the attributes related with the diseases/disorders identified by cTAKES we found information that could be directly used for some of the modifiers: we used the polarity attribute from cTAKES to identify if the diseases/disorders were negated and assigning a value to `NI`; for the `SC`, `UI` and `CO` modifiers, cTAKES have attributes with the same name and we only needed to convert that information into the normalized values of the task modifiers.

For the `BL` modifier we used a set of rules to know if there were body locations in the same sentence of the identified disease/disorders and extracted the respective CUI. We tried to extract the CUI of the most specific body location possible, so we searched the expression with a bigger number of words, using the premise that more information means more specificity.

Course Class and Severity Class. For the `CC` and `SV` modifiers we used a mapping approach. From the 299 clinical texts that compose the training corpus, we extracted expressions (without repetition) related to each modifier value.

When using expressions from a mapping approach, there is the risk of identifying equal expressions from the text but not in the correct context. To determine if the modifiers `CC` and `SV` had this problem we checked the expressions in each mappings file and concluded that the expressions were not too common and the probability of identifying wrong expressions was acceptable for our objectives.

Generic Class. The `GC` modifier had a particular characteristic – there was no example of it in the training corpus; assuming that the test corpus would follow this, few to none appearances of this modifier expressions would appear. Based on this assumption we used the default value (`false`) in every template.

[2] `http://ctakes.apache.org/`

DocTime. The `DT` modifier expresses the temporal relation between the disease/disorder and the time when the clinical text was written. It can have the following values:

- `Before-overlaps`: disease/disorder identified in the past and still present;
- `Before`: disease/disorder identified and treated in the past;
- `Overlap`: disease/disorder present but there is no information about when it was diagnosed or when it will pass;
- `After`: one action or event that it is still to come;
- `Unknown`: no temporal relation information.

For this modifier we used a purely statistic approach, meaning that, for each template we selected the most common value presented in the training corpus – `Overlap`.

Table 3 presents occurrence percentage for training corpus for each possible `DT` value; it can be noticed that more than half of the occurrences (56.35%) has the `Overlap` value, so this one was chosen to fill all the templates. The `Before` value had also an expressive number, but `Overlap` more than doubles it.

Table 3. `DocTime` values distribution in the training corpus.

Value	no. occurrences	%
Before-overlaps	2814	16.41
Before	4205	24.52
Overlap	9666	56.35
After	442	2.58
Unknow	25	0.14
Total	17152	100.00

Temporal Expressions. To identify dates and hours we used regular expressions. At first we thought of using a mapping approach too, but dates and hours are very specific and if an expression appear in the same format but with one day apart, that expression wouldn't be identified.

Based in the training corpus, we created four regular expressions aiming to identify `DATE` and two regular expressions to identify `Time`:

- DATE
  - Day/Month/Year (dd/mm/yyyy);
  - Day-Month-Year (dd-mm-yyyy);
  - Year-Month-Day (yyyy-mm-dd);
  - Month-Year (mm-yy).
- TIME
  - 24 hours time (hh:mm);
  - 12 hours time (hh:mm am/pm)

We didn't consider the identification of expressions associated with the remaining values of the modifier – duration and set.

3.3 Implementation

Our system was implemented using the Java programming language. Figure 1 presents the system's architecture – it uses mapping files, regular expressions, decisions based on the training corpus and cTAKES.

XML files were generated from cTAKES, and from them we extracted information using a parser and applied the procedures described in the last subsection. With the obtained information, the system updated the modifiers' values and printed the templates with the final result.

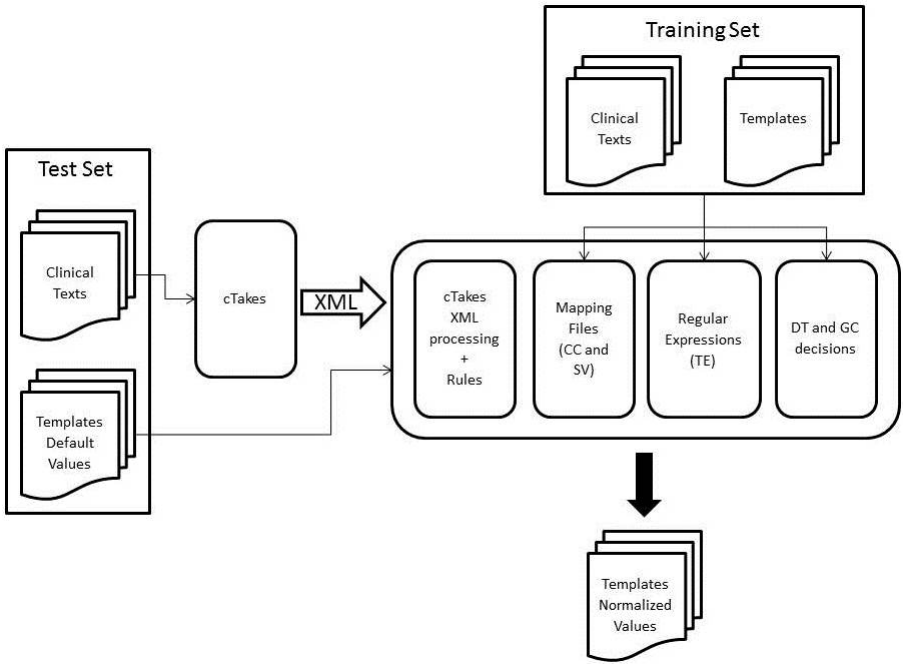

Fig. 1. System description

Next we explain the steps necessary to get the filled templates:

1. run cTAKES with the clinical texts as input;
2. load information from templates, namely the header (because the rest are the default values), and the map files built for `CC` and `SV`;
3. process the XML from cTAKES using a set of rules to extract information;
4. use the information previously gathered to substitute the default values from the templates.

Step 1. The first step can be also called a pre-processing one – the generation of the XML files using cTAKES. It generates a XML file for each clinical text.

cTAKES has a large set of specific analysis engines and a set of aggregate ones that combine the specific ones. These aggregate engines describe how particular annotators can be combined using a set of rules that describe how each annotator uses the analysis of the previous one.

Several aggregate engines were tested and the one that offered the best results (and was used for the participation run) was `AggregatePlaintextUMLSProcessor`.

Step 2. On startup, the system loads the mapping files of `CC` and `SV` modifiers obtained from the training corpus. It also loads the templates information into a data structure that the system can use during all process.

Step 3. After steps 1 and 2, the system processes the XML files. We used `xPath` expressions to extract the information considered necessary to task 2a; this information was stored in data structures suited for being subsequently processed. The information is extracted using two approaches:

- the 'strict' one, where the system searches diseases/disorders with a perfect match the information gathered from cTAKES;
- the 'relaxed' one, that is used in case the 'strict' fails. This one, although less accurate, verifies if the boundaries of the disease/disorder from the template header are inside the ones of the chunk identified by cTAKES.

The CUI of the body locations associated to the disease/disorder is obtained using a set of rules that joins information from the different data structures mantained. In order to reach the most specific CUI, the system chooses the longest body location term from the cTAKES output.

Step 4. The final step gathers all information from the previous steps, relying mainly in the coordinates of the diseases/disorders in text.

To extract the modifiers information, the system searches the sentences where the diseases/disorders were identified, looks for the cTAKES gathered information, replaces the info in the respective template, searches for terms in the mapping, applies the regular expressions and writes the found info in the template. Finally it writes the info for the `DT` and `GC` modifiers (that is equal for all templates).

4 Results

Table 4 presents the accuracy obtained by our system for the train and test corpora, and also the best accuracy obtained for each modifier in the task 2a.

Analysing the table we see that the overall accuracy between the train and test corpora have a difference less than 0.03. For most of the modifiers the accuracy between the train and the test corpora don't differ more than 0.02, but in some of them the test corpus's accuracy is better: `BL` has an improvement of

Table 4. System's accuracy for the train and test corpora and the best accuracy reported on task 2a for each modifier.

modifier	train	test	best
NI	0.916	0.901	0.969
SC	0.991	0.987	0.995
UI	0.932	0.955	0.960
CC	0.866	0.859	0.971
SV	0.915	0.919	0.982
CO	0.978	0.975	0.978
GC	1.000	1.000	1.000
BL	0.469	0.540	0.797
DT	0.59	0.024	0.328
TE	0.715	0.857	0.864
Overall	0.837	0.802	0.884

0.071 and `TE` an improvement of 0.142. For `DT` modifier, the training presents a better result with an improvement of 0.57 over the test corpus.

Comparing the test corpus results with the best accuracy reported in task 2a we notice that in some modifiers like `SC`, `UI`, `CO` and `TE` the difference is lower than 0.2 and the values for class `GC` are equal; for modifiers `BL`, `DT` and `CC` there is a bigger discrepancy between the results. Nevertheless, in overall our system stood behind 0.082 when compared with the overall value calculated.

Table 5 presents the F_1, precision and recall values for both the train and test corpora. There we can see that the values are not so different between the train and test corpora among most of the modifiers. Modifiers like `SC`, `UI`, `CO`, `BL` and `TE` have better results in the test corpus; on the other side `NI`, `CC`, `SV` and `DT` modifiers have better results in the training corpus.

Table 5. F_1, precision and recall for training corpus and test corpus.

modifier	train			test		
	F_1	precision	recall	F_1	precision	recall
NI	0.744	0.914	0.628	0.723	0.862	0.622
SC	0.495	0.408	0.631	0.556	0.532	0.581
UI	0.409	0.886	0.266	0.451	0.813	0.312
CC	0.385	0.257	0.771	0.264	0.165	0.661
SV	0.670	0.546	0.868	0.547	0.400	0.866
CO	0.723	0.942	0.587	0.760	0.955	0.631
GC	0	0	0	0	0	0
BL	0.232	0.255	0.213	0.253	0.265	0.243
DT	0.592	0.590	0.593	0.024	0.024	0.024
TE	0.155	0.581	0.089	0.233	0.425	0.161
Overall	0.479	0.513	0.448	0.214	0.217	0.211

The DT modifier obtained widely different values with a F_1 of 0.592 in the train and a corresponding value of 0.024 in the test corpus. This can be explained because the value of this modifier is always the same for every template of the output; this decision was based on the modifier statistics from the training corpus.

We ranked seventh among all the participants of task 2a, as showed in Table 6. The best system had an overall accuracy of 0.868 and our system obtained an overall accuracy of 0.802. This value is lower than the average accuracy value of all participants. Our system also obtained values below the average in the F_1, precision and recall.

Table 6. Relative performance for task 2a.

system	accuracy	F_1	precision	recall
TeamUEvora (rank 7)	0.802	0.214	0.217	0.211
Best system	0.868	0.499	0.485	0.514
Average	0.814	0.273	0.308	0.269

Table 7 shows the relative performance of full template accuracy of our system, the best value obtained and the average of all participants. The best value is below 0.2 and our system obtained a very low value of 0.007.

Table 7. Relative performance for task 2a of full template accuracy.

system	accuracy
TeamUEvora (rank 11)	0.007
Best System	0.196
Average	0.056

5 Conclusions and Future work

This paper presents the design and the implementation of our system, developed for participating in the task 2a of 2014 ShARe/CLEF eHealth Lab. The task's main goal was to obtain normalized attributes values for disease/disorder template filling.

5.1 Conclusions

Our participation's main goal was to understand the problems associated with the design and implementation of a system to extract information from medical

data. The system gathers knowledge from already implemented technology in the clinical area, namely cTAKES; it also uses resources based on the training corpus, regular expressions and decisions based on modifiers statistics.

Between 14 participants, it ranked 7th, with an accuracy value of 0.802. Taking into account our goal, we consider this a good result; nevertheless there is much space for improvement.

5.2 Future work

cTakes is one of the resources of our system and we intend to add more sources of knowledge of the medical area so we can improve our system. One hypothesis is MetaMap[5], widely used in task 1 of 2013 Lab. Last year, some participants used only cTAKES [6,8], others used only MetaMap [7,9,10,11,12] and others used a joint approach [13,14].

On the other hand, we intend to complement or substitute the approach taken to some modifiers:

- for `Course` and `Severity` we want to try a machine learning approach;
- for temporal expressions, we want to improve the system by also identifying duration and set expressions. For that we intend to use technologies in the area of clinical time identification;
- for `DocTime` we intend to incorporate knowledge in order to give different values to different examples (instead of using the same value for all of them):
- for `Generic` modifier, we aim to develop a more automatic way to detect this class. Nevertheless, to do that we need some examples of this modifier in the training corpus.

References

1. Kelly, L., Goeuriot, L., Leroy, G., Suominen, H., Schreck, T., Mowery D. L., Velupillai, S., Chapman, W. W., Zuccon, G., Palotti, J.: Overview of the ShARe/CLEF eHealth Evaluation Lab 2014. Springer-Verlag.(2014)
2. Elhadad, N., Chapman, W., O'Gorman, T., Palmer, M., Savova, G.: The ShARe Schema for the Syntactic and Semantic Annotation of Clinical Texts. (2014). (Under Review).
3. Suominen, H., Salanterä, S., Velupillai, S., Chapman, W. W., Savova, G., Elhadad, N., Pradhan, S., South, B. R., Mowery, D. L., Jones, G. J., Leveling, J., Kelly, L., Goeuriot, L., Martinez, D., Zuccon, G.: Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. CLEF 2013, Valencia, Spain: Springer Berlin Heidelberg. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
4. Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Kipper-Schuler, K.C., Chute, C.G.: Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. In: Journal of the American Medical Informatics Association 17 (2010) 507-513.
5. Aronson, A.R., Lang, F.M.: An overview of MetaMap: historical perspective and recent advances. JAMIA 17(3) (2010) 229-236.

6. Cogley, J., Stokes, N., Carthy, J.: Medical Disorder Recognition with Structural Support Vector Machines. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
7. Leaman, R., Khare, R., Lu, Z.: NCBI at 2013 ShARe/CLEF eHealth Shared Task: Disorder Normalization in Clinical Notes with DNorm. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
8. Gung, J.: Using Relations for Identication and Normalization of Disorders: Team CLEAR in the ShARe/CLEF 2013 eHealth Evaluation Lab. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
9. Hervás, L., Martínez, V., Sánchez, I., Díaz, A.: UCM at CLEF eHealth 2013 Shared Task1. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
10. Osborne, J. D., Gyawali, B., Solorio, T.: Evaluation of YTEX and MetaMap for clinical concept recognition. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
11. Wang, C., Akella, R.: UCSC's System for CLEF eHealth 2013 Task 1. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
12. Zuccon, G., Holloway, A., Koopman, B., Nguyen A.: Identify Disorders in Health Records using Conditional Random Fields and Metamap; AEHRC at ShARe/CLEF 2013 eHealth Evaluation Lab Task 1. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
13. Bodnari, A., Deleger, L., Lavergne, T., Neveol, A., Zweigenbaum, P.:A Supervised Named-Entity Extraction System for Medical Text. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).
14. Xia, Y., Zhong, X., Liu, P., Tan, C., Na, S., Hu, Q., Huang, Y.: Combining MetaMap and cTAKES in Disorder Recognition: THCIB at CLEF eHealth Lab 2013 Task 1. In: Proceedings of ShARe/CLEF eHealth Evaluation Labs (2013).