Optimizing Apache cTAKES for Disease/Disorder Template Filling: Team HITACHI in 2014 ShARe/CLEF eHealth Evaluation Lab

[1]Nishikant Johri, [2]Yoshiki Niwa, and [3]Veera Raghavendra Chikka

[1]Research and Development Centre, Hitachi India Pvt Ltd, Bangalore, India
[2]Hitachi, Ltd., Central Research Laboratory, Japan
[3]International Institute of Information Technology, Hyderabad, India
nishikant@hitachi.co.in,yoshiki.niwa.tx@hitachi.com,
raghavendra.ch@research.iiit.ac.in

Abstract. This paper describes an information extraction system developed by team Hitachi for "*Disease/Disorder Template filling*" task organized by ShARe/CLEF eHealth Evaluation Lab 2014. We approached the task by building a baseline system using Apache cTAKES. We submitted two separate runs; in our first run, rule based assertion module predict the norm slot value of assertion attributes excluding training data knowledge. However assertion module is changed to machine learning-based in second run. We trained models for Course modifiers, Severity modifier and Body Location relation extractor and applied a variety of rule based post processing including structural parsing. We performed two layer search on UMLS dictionary for refinement of body location. Eventually, we created rules for temporal expression extraction and also used them as features for model training of DocTime. We followed a dictionary matching technique for cue slot value detection in Task 2b. Evaluation result of test data showed that our system performed very well in both subtasks. We achieved the highest accuracy 0.868 in norm value detection, strict F1-score 0.576 and relaxed F1-score 0.724 in cue slot value identification, indicating promising enhancement on baseline system.

Keywords: Natural language processing, information extraction, Apache cTAKES, UMLS, relation extraction, dictionary matching, CRF, SVM, rule based assertion, structural parsing

1 Introduction

With the widespread usage of electronic health record (EHR), a large amount of healthcare data is being generated, posing massive challenges in doing effective analysis for stakeholders (administrators, care providers and researchers) and foreshadowing text mining as one of the dominant fields of research in medical domain. The adoption of natural language processing (NLP) in healthcare has opened a door for patients' better understanding on their health and paved the way for advanced research in medical field.

For the past few years, numerous healthcare research organizations have focused their efforts toward unraveling the enigmatic nature of clinical text and promoting research in medical domain. In this series, ShARe/CLEF eHealth Evaluation Lab 2013 introduced three challenging tasks on NLP and information retrieval (IR) and extended them to ShARe/CLEF eHealth Evaluation Lab 2014 in which we submitted our system for information extraction task.

1.1 Task Description

Task 2 "*Information extraction from clinical text: Disease/Disorder Template Filling*" in CLEF eHealth 2014 is an extension of CLEF eHealth 2013 task "*Named entity recognition and normalization of disorders*" [1].

In template filling task, the participants have been provided with a corpus of de-identified healthcare reports along with an empty template for each disease/disorder mentioned in the report. The template consists of mention's UMLS [2] CUI, span offset of mention and a list of unfilled value slots for 10 attributes: Negation Indicator, Subject Class, Uncertainty Indicator, Course Class, Severity Class, Conditional Class, Generic Class, Body Location, DocTime Class, and Temporal Expression. The participants have to prepare a system which can predict the norm value for each attribute value slot from a list of possible norm values. An optional task on cue slot value identification (span offset of lexical cue) is also conducted in which participants are asked to find the span offset of each attribute from healthcare reports.

1.2 Corpus Description

As the task is an extension of CLEF eHealth 2013 tasks, the resulting dataset of CLEF eHealth 2013 Task 1 and Task 2 has been served as training corpus for system development in CLEF eHealth 2014. The training corpus consists of 4 types of healthcare reports: Discharge summary, Radiology report, ECHO report and ECG report, while test data has only Discharge summaries. Table 1 describes corpus statistics.

Table 1. Statistics of training and test data

Report type	**Training dataset**		**Test dataset**	
	#reports	#annotations	#reports	#annotations
DISCHARGE SUMMARY	136	9098	133	8003
RADIOLOGY REPORT	54	831	0	0
ECHO REPORT	54	1429	0	0
ECG REPORT	54	196	0	0

2 Processing Pipeline and Approaches

We approached the task by building a baseline system using Apache clinical Text Analysis and Knowledge Extraction System (cTAKES) [3, 4]. Although few cTAKES modules are still under development, we followed its clinical pipeline for development of baseline system.

2.1 System Architecture

Apache cTAKES is a NLP framework specifically built for processing medical text. Figure 1 depicts our system architecture built upon cTAKES. It takes clinical text as input, applies cTAKES preprocessing steps followed by individual module's process and finally the generated result is supplied to Template Filler which ultimately resolves norm value and identifies cue slot value of attributes.

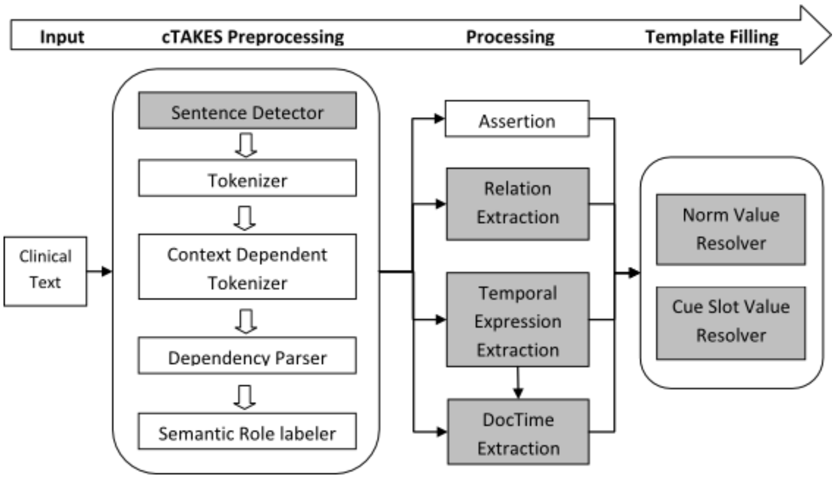

Fig. 1. System Architecture and Processing Pipeline built upon Apache cTAKES. The grey components are modified or rebuilt while other components remain unchanged from original cTAKES framework.

2.2 cTAKES Preprocessing

Our system relies on cTAKES for preprocessing of clinical text. We analyzed the training corpus and modified sentence detector module to overcome the problem of fallible end of sentence. We merged two or more sentences which erroneously cover a single disease/disorder mention. On the other hand, we overcome the disjoint disease/disorder problem by using last text span as target disease/disorder because in most of the disjoint training examples last text span represents disease/disorder and consequently it can predict attribute norm values too. For example, in the disjoint span "*left atrium ... dilated*", *dilated* is used as target disease/disorder.

The majority of preprocessing is accomplished using cTAKES components. Sentence detection is followed by tokenization, part of speech tagging and NP-chunking. Tokenization is followed by context dependent tokenization which classifies tokens into various categories such as number-token, date-token, time-token, range-token etc. Finally, dependency parser and semantic role labeler (SRL) are applied over tokenized data providing dependency relationship between semantic arguments.

2.3 Processing Individual Modules

2.3.1 Assertion: Our system uses cTAKES assertion module to determine norm value of Negation Indicator, Uncertainty Indicator, Condition Class, Subject Class and Generic Class. Assertion attributes can be determined using machine learning as well as rule based approach. We submitted separate runs for both the approaches.

Rule Based Assertion: In rule based assertion, NegEx [5] algorithm has been used to predict whether the mentioned disease/disorder is negated, more specifically it resolves the polarity of sentence. It requires predefined negation phrases which have been divided into two groups *pseudo-negation* and *actual-negation.* Pseudo negation consists of phrases that appear to indicate negation but instead identify double negative ("*not ruled out*") and ambiguous phrasing ("*unremarkable*"). For instance, in the sentence "*Ambulating without difficulty, chest pain free, and without further evidence of bleeding*", all diseases *ambulating*, *chest pain* and *bleeding* are negated by pseudo negation phrases *without difficulty*, *free* and *without further evidence of* respectively. On the other hand, actual negation phrase denies disease or finding when used in a window of +/- 5 tokens including stop words. For example, in the sentence "*Ext: No clubbing, cyanosis or edema*" all the findings are negated by *No* phrase.

The Assertion module predicts uncertainty and conditional class by scoring the target phrase using list of words with predefined uncertainty/conditional values. For example *if*, *risk*, *evaluate*, *when*, *check* are some typical words with high conditional score. Similarly, *uncertain*, *differentiate*, *suspect* are tokens having high uncertainty score. For scoring, it also uses POS tags and token entries present in the left window of mentioned disease.

We used cTAKES feature selection for subject class identification. The feature set includes token, SRL argument, dependency path and SRL dependent token of all the persons who appeared in the sentence mentioning disease/disorder. A rule based approach is applied over the selected features. For example, if system does not find any feature, the subject is patient. If donor and family member both features are true, the subject will be *donor_family_member*. Similarly other cases have been introduced to predict the subject experiencing disease/disorder.

As per the training dataset, there is no entry asserting generic attribute in whole dataset. However, we used generic classifier of assertion module to classify the generic attribute.

Using Assertion models: In machine learning method, we used ClearTK [6] framework for feature extraction and trained separate models for each assertion attribute on training data.

Apache cTAKES provides method for feature selection for assertion attribute. All assertion attributes have a common feature list which includes word, word-stem, tokens within -/+ 5 window and bag of words within -/+ 3 window of disease/disorder mention. An additional feature word score is derived by taking mean of contextual token distance from the mentioned disease/disorder.

For each attribute, some additional features are added along with the common feature list. Negation dependency features are used for polarity detection. For subject class, all features of rule based approach along with outcomes are used in training subject class model. Unlike negation indicator and subject class no additional features are used for training uncertainty and conditional class models. In contrast to other assertion attributes, generic attribute does not have any positive classification in training data; however it does have cue slot value which made it easier to prepare generic classifier. Generic model also employed features and outcomes derived from rule based approach.

Assertion cue slot identification: For assertion attributes, we followed a dictionary matching approach for cue slot identification. We created stem dictionaries from training data comprising stems of attribute's keywords. Table 2 shows sample stem dictionaries of assertion attributes.

Table 2. Sample stem dictionary for assertion attributes

Subject	**Negation Indicator**	**Uncertainty**	**Conditional**	**Generic**
father	no evidence of	differentia	if	recommended
mother	no sign of	uncertain	Concern	consult
famil	negative	potential	protect	sign
parent	absent	probab	when	service
paternal	without	suspec	indicat	mention

Dictionary matching is performed on the sentence mentioning disease/disorder. Negation indicator's cue slot value is determined by extracting nearest negation phase present in the left window of mentioned disease/disorder. However complete window is considered when extracting uncertainty, conditional and generic class cue slots. In contrast to other attributes, subject class cue slot is identified using dictionary matching over SRL arguments.

2.3.2 Relation Extraction: Apache cTAKES treats the task of locating body sites and severity modifier as a relation extraction problem [7]. When handling body location finding, we evaluated the results of cTAKES relation extractor and found scope of improvement by applying rule based post processing; however because of promising results of severity and course class, our system completely relied on relation extractor for severity and course class identification.

Body Location Extraction: Body Location is the most critical attribute in template filling task concerning CUI ambiguity of clinical concepts. Figure 2 depicts a typical sequence of algorithms applied for body location finding.

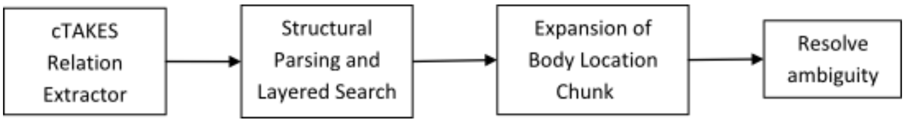
cTAKES Relation Extractor
Structural Parsing and Layered Search
Expansion of Body Location Chunk
Resolve ambiguity

Fig. 2. Sequence of algorithms applied for body location finding

As a preprocessing step in body location identification, we built a lucene [8] based dictionary comprising all UMLS concepts of body parts falling into different semantic types defined by CLEF eHealth 2014 [9]. We indexed first term as well as full text of concept in order to implement a layered search approach.

In first layer of search, dictionary lookup is applied over NP chunks, providing entity mention annotations (body part, anatomical sites and their associated CUIs). After annotating entity mentions, features are generated for all possible pairs of disease and entity mentions. For example, in the sentence "*patient has severe pain in left shoulder while right shoulder is normal*", *pain* is disease and *left shoulder* and *right shoulder* are two body locations. In this case, two training instances *pain .. left shoulder* and *pain .. right shoulder* (true and false respectively) are generated to find the relationship between arguments. We trained support vector machine (SVM) model for *location_of* relation extractor using cTAKES relation extractor module and training dataset. However machine learning method failed to identify some body location relations. For instance, in the sentence "*Ext: trace bilateral lower ext edema; R groin small hematoma, no bruits*", *Ext* is an abbreviation of Extremities (body location) but machine learning model could not detect it.

In order to find the missing body locations, we apply a second layer of search enabling structural parsing which determines body location in sections, subsections and headings of the document especially in ECHO REPORT and DISCHARGE SUMMARY. Section headings play an important role when relating disease with the body location especially in section containing very short/long sentences such as "*Neck: No JVD. JVP 7 cm. No carotid bruits*". Here *Neck* is the target body location for disease *JVD*, *JVP* and *carotid bruits*. Following are few sentences where section heading determines relationship.

- *MITRAL VALVE: The mitral valve leaflets are mildly thickened. There is mild mitral annular calcification. Mild (1+) mitral regurgitation is seen.*
- *AORTIC VALVE: Normal aortic valve leaflets (3). No AS. No AR.*
- *Cardiac: RRR. S1/S2. No M/R/G*
- *Extremities: No C/C/E bilaterally, 2+ radial, DP and PT pulses b/l.*

Another missing body location case is when body location is present within disease/disorder mention itself. For example, the sentence "*There is a mild mitral annular calcification*" has *mitral annular calcification* as disease and *mitral annular* as target body location which machine learning system could not detect in many instances. Therefore, in second layer search, we find body locations within mentioned disease/disorder.

After extracting relationship between body part and disease, we expand body part text chunks to +/- 5 token windows. For example, in the sentence *EGD showed food compaction in the lower third of the esophagus and gastroesophageal junction*, first layer search finds *esophagus* which is expanded to *lower third of the esophagus* and hence overlapping results are transformed into strict results resolving CUI ambiguity to some extent.

Finally the problem of ambiguity is tackled. For example, UMLS search on *Sinus* gives CUI C1305231, but in ECG reports *Sinus* implicitly refers to *Coronary Sinus* (C0456944). Similarly, in sentence "*intact in all four extremities to LT, PP, cold*", *extremities* CUIs are C0015385 and C0278454 but the correct CUI is C0278454. In order to resolve ambiguity, we prepared separate dictionaries for each report type. Each dictionary includes anatomical sites pertaining to the report type. For example, ECG dictionary includes heart, chambers and other heart components. However, dictionaries have been created from supplied training data and CUI ambiguity has been resolved by considering most frequent CUI.

Severity and Course Class: cTAKES relation extractor provides *degree_of* relation which enlightens the degree to which a disease/disorder is modified. Using cTAKES modifier extractor and supplied training data, we prepared two separate conditional random field (CRF) machine learning models for severity and course modifier. For CRF training, feature set contains only tokens covered and POS tags. After annotating severity modifiers using CRF model, features are generated for all possible pairs of disease mention and severity modifiers and similar to body location relation extraction, SVM models are trained for *degree_of* relation.

Once relationship is determined, normalization of modifier is approached using synonym stem dictionaries. We collected all the nearest synonyms of norm values and prepared a dictionary comprising stem of severity modifiers and their synonyms present in training data. The same approach has been followed for course class identification and normalization. Table 3 shows sample stem dictionaries of severity and course modifiers.

Table 3. Sample list of severity and course modifiers with synonym stems

Modifier type	Class	Synonym stems
Severity Modifier	severe	advanc, bad, dart, elevat,
	slight	small, little, minimal, niggl
	moderate	check, control, mild, moderat
Course Modifier	increased	increas, high, advanc, ascend, addition
	improved	improv, normal, better, come-back, well
	resolved	recover, regain, block-up, ceas, clear
	decreased	decreas, contract, declin, degenerat, dim
	changed	chang, evolv, moving, transform
	worsened	worse, spoil, swell, tough, wretch

2.3.3 Temporal Expression Extraction: We approached temporal expression finding with a rule based technique. Most of the temporal expression in ECHO, ECG and RADIOLOGY reports are taken either from document header or from DATE/TIME heading. For example, in the following header of ECG report, '2016-01-05'(DATE) is the temporal expression which has been extracted during structural parsing of document.

- 83601||||1114||||23168||||ECG_REPORT||||2016-01-05 03:57:00.0|||| |||| |||| ||||

Similarly, temporal expression has been extracted from heading "Date/Time:" in ECHO report and from "DATE:" in radiology report.

On the other hand, discharge summaries include more critical temporal patterns. We built finite state machines (FSM) for numerical date and time patterns. Besides FSM, we also developed an algorithm to find textual temporal expressions.

The algorithm divides all the temporal keywords into various classes. Table 4 shows typical keywords representing time. We created patterns shown in Table 5, which can match non-space separated time expressions such as *1day before*, *hd2*, *x1 yr*, *5am*, *12p.m* etc. Another Table 6 contains adjuster and modifier that usually occurs before and after the time keywords. All other words are considered in NONE category including stop-words.

The algorithm first generates temporal equivalence of sentence, mapping each token to one of the classes listed in Table 4, Table 5 or Table 6. It then looks up for chunks having class of Table 4 and Table 5 and expands them to left and right window by using adjusters and modifiers in Table 6 until two adjacent NONE or stop-word appear. For example, in the sentence "*On the evening of postoperative day three, the patient had another short 7-beat run of ventricular*

Table 4. Time classes and keywords

Time Class	Keywords
Unit	second, month, week, year, decade, century, y, m, d etc
PartOfDay	morning, afternoon, evening, night, overnight
DayOfWeek	monday, tuesday, mon, tue, wed etc
MonthOfYear	january, february, march, jan, feb, mar etc
SeasonOfYear	spring, summer, fall, autumn, winter
Time	now, today, tonight, yesterday, noon, a.m
Duration	times, duration, interval, x
Date	hd, pod

Table 5. Derived time classes and regular expression

Derived Time Classes	Regular Expression
INT_ROMAN	\d+\B(st|nd|rd|th)\b
DUR_UNIT	\d+\B(UNIT)(DURATION)\b
TIME_UNIT	\d+\B(UNIT)(TIME)\b
DATE_UNIT	\b(DATE)\B\d+

Table 6. Adjuster and Modifier keywords

Adjuster	Keywords
Number	All integers, one, two, three, twenty, thirty, hundred etc
TimeReference	previous, previously, recent, recently etc
Frequency	every, each, hourly, daily, frequently
Adjuster	last, past, previous, ago, next, prior, throughout
Modifier	few, half, within
PrePost	preoperative, postoperative, preop, postop, pre-surgical,

tachycardia", *evening* has class PartOfDay and *day* has Unit, so it is expanded to *evening of postoperative day three* because next two tokens are NONE or stopword. Furthermore, we relate disease/disorder to the nearest temporal expression when multiple temporal expressions are found in a sentence.

Once temporal expression is found, it has to be classified into one of the three norm values DATE, TIME or DURATION. Table 7 shows the classes and corresponding dictionary categories and keywords.

Table 7. Temporal expression class, their categories and keywords

Class	Categories	Keywords
DURATION	DUR_UNIT, Duration, SeasonOfYear,	year, month, day, week, year, wk , period, century, Past, over, within, since, throughout, through, several
TIME	TIME_UNIT, PartOfDay, TimeAnnotation	ago, before, after, prior, earlier, hour, min, sec, am, pm
DATE	Prepost, DATE_UNIT, Date, MonthOfYear, Year, INT_ROMAN, DayOfWeek, DateAnnoation	postoperative, pod, day, date

2.3.4 DocTime Extraction: DocTime class indicates temporal relation between a disease/disorder and document authoring time. We used cTAKES DocTime module with some enhancement of feature selection. The feature set included in DocTime module contains tokens and POS tags within +/-3 window of mentioned disease/disorder, tense of nearby verb, section heading and closest verb. Along with these features, we also integrated time expression features found during temporal expression extraction phase.

3 Evaluation

Our system was developed on a training set (298 reports) and evaluated on a test set (133 reports) supplied by the organizer. All machine learning models are optimized using 10-fold cross validation on the training data; however no additional annotations are used throughout the development.

3.1 Evaluation metric

According to the organizer's evaluation criteria, evaluation focuses on accuracy for Task 2a (norm value detection) and F1-score for Task 2b (cue slot identification), defined as strict F1-score (span is identical to the reference standard span) and relaxed F1-score (span overlaps reference standard span). Each task has been evaluated by overall performance as well as attribute type.

3.2 Results

As reported by the organizer, our system achieved the best results in both of the information extraction tasks: Task 2a (norm value detection) and Task 2b (cue slot value identification). Table 8 shows overall performance of our system in Task 2a and Task 2b. Table 9 shows per attribute type result for both tasks.

Table 8. Overall performance of our system in Task 2a and Task 2b

Task	**System**	**Overall Result**			
		Accuracy	**F1-score**	**Precision**	**Recall**
(2a)	TeamHITACHI.2	0.868	0.499	0.485	0.514
	TeamHITACHI.1	0.854	0.478	0.453	0.506
(2b) (Strict)	TeamHITACHI.2		0.576	0.535	0.624
	TeamHITACHI.1		0.573	0.535	0.616
(2b) (Relaxed)	TeamHITACHI.2		0.724	0.672	0.784
	TeamHITACHI.1		0.719	0.672	0.773

Table 9. Per attribute type result for Task 2a and Task 2b

Attribute type	**Task 2a**	**Task 2b**					
	Accuracy	**F1-score**		**Precision**		**Recall**	
		Strict	**Relax**	**Strict**	**Relax**	**Strict**	**Relax**
Body Location	0.797	0.735	0.874	0.754	0.897	0.717	0.853
Course Class	0.971	0.6	0.67	0.567	0.632	0.638	0.712
Conditional Class	0.978	0.352	0.801	0.382	0.869	0.326	0.743
DocTime Class	0.328						
Generic Class	0.99	0.203	0.304	0.213	0.32	0.193	0.289
Negation Indicator	0.969	0.775	0.926	0.804	0.962	0.747	0.893
Subject Class	0.993	0.119	0.165	0.066	0.092	0.589	0.814
Severity Class	0.982	0.828	0.85	0.836	0.857	0.821	0.843
Temporal Expression	0.773	0.239	0.37	0.201	0.31	0.297	0.458
Uncertainty Indicator	0.96	0.419	0.672	0.381	0.612	0.465	0.746

3.3 Discussion

As shown by the evaluation results, our system outperformed not only in overall evaluation but also in majority of attribute type evaluations; however measures also insinuate the scope of improvement in our system.

In Task 2a, our system (0.868) achieved an improvement of 0.025 in overall accuracy benchmarking with respect to second best team (0.843). We submitted two runs, TeamHITACHI.1 employed rule based assertion while TeamHITACHI.2 predicted assertion attribute using machine learning method. Result in Table 8 shows that machine learning method (0.868) improved the results by 0.014 in comparison to rule based assertion system (0.854). On the other hand, in Task 2b, our system's performance is best among all the submitted systems; however low F1-score clearly implies the scope of improvement in cue slot identification.

According to the attribute wise evaluation, our system obtained the highest accuracy in 7 out of 10 attributes in Task 2a. As body location was the most critical attribute, our system achieved the highest accuracy 0.797 in mapping body location to UMLS CUI. The difference between strict F1-score 0.735 and relaxed F1-score 0.874 for body location cue slot identification suggests amendment of dictionaries and optimization of dictionary lookup algorithm. Another concerning attribute which achieved least accuracy 0.328 is DocTime class, albeit highest among all the systems. One possible feature enhancement for DocTime relation could be inclusion of features other than sentence feature because when a disease is described in more than one sections, all section's information contribute in prediction. For example, in the document "*He has a history of schizoaffective disorder and anxiety..... Schizoaffective disorder: restarted psychiatric medications once he was awake enough to eat.*", *schizoaffective disorder* is patient's history as well as current problem, but *anxiety* has history only, therefore, *schizoaffective disorder* should be assigned BEFORE_OVERLAP and *anxiety* should be assigned BEFORE DocTime class. At last, our rule based temporal expression extractor did not perform well (accuracy 0.773) and ranked 9th in the normalization task, indicating refinement of temporal rules.

For Task 2b, we deliberately focused on relaxed F1-score in cue slot value identification, consequently the overall relaxed F1-score (0.724) exceeded the overall strict F1-score (0.576). The low F1-score of assertion attributes insinuate ineffectiveness of dictionary matching method for assertion attribute tagging. It will remain an open ended problem for future works.

4 Conclusions

The paper described our method and pipeline employed to fill the disease/disorder template in our submission to Task 2 of ShARe/CLEF eHealth Evaluation Lab 2014. We began with the baseline system development using Apache cTAKES and incorporated many cTAKES modules in our system. We developed several wrappers comprising machine learning and rule based techniques for norm value detection of various attributes. We performed rule based post processing including dictionary matching to identify attribute's cue slot value. The evaluation results demonstrated that our system achieved the best accuracy in both norm and cue slot value identification task, indicating promising enhancement over baseline system. However the results also signify the scope of improvement in some modules, especially in cue slot identification.

Acknowledgements

We would like to thank International Institute of Information Technology, Hyderabad India for providing us technical assistance through survey of existing technologies and implementation of severity and course class dictionaries. We are also very thankful to task organizers of CLEF eHealth 2014, who were funded by the United States National Institutes of Health.

References

1. H. Suominen, S. Salantera, S. Sanna, and et al, "*Overview of the ShARe/CLEF eHealth Evaluation Lab 2013*", presented at the Proceedings of CLEF 2013, 2013.
2. UMLS (Unified Medical Language System) `http://www.nlm.nih.gov/research/umls/`
3. G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, "*Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications*", J Am Med Inform Assoc, vol. 17, no. 5, pp. 507âĂŞ513, Sep. 2010.
4. Apache cTAKES (clinical Text Analysis and Knowledge Extraction System) `http://ctakes.apache.org/`
5. Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper and Bruce G. Buchanan, "*A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries*", at Journal of Biomedical Informatics 34, 301-310 (2001).
6. P. V. Ogren, P. G. Wetzler, and S. J. Bethard. "*ClearTK: a framework for statistical natural language processing*", Unstructured Information Management Architecture Workshop at the Conference of the German Society for Computational Linguistics and Language Technology, 9 2009.
7. Dmitriy Dligach, Steven Bethard, Lee Becker, Timothy Miller, and Guergana K Savova, "*Discovering body site and severity modifiers in clinical texts*" J Am Med Inform Assoc 2013.
8. Apache Lucene `http://lucene.apache.org/core/`
9. L Kelly, L Goeuriot, G Leroy, H Suominen, T Schreck, DL Mowery, S Velupillai, WW Chapman, G Zuccon, J Palotti. "*Overview of the ShARe/CLEF eHealth Evaluation Lab 2014.*" Springer-Verlag.
10. N Elhadad, W Chapman, T O'Gorman, M Palmer, G Savova. "*The ShARe Schema for the Syntactic and Semantic Annotation of Clinical Texts.*" Under Review.