ShARe/CLEFeHealth: A Hybrid Approach for Task 2

Huu Nghia Huynh, Son Lam Vu and Bao Quoc Ho

Faculty of Information Technology
University of Science, HoChiMinh City, VietNam
huynhnghiavn@gmail.com, lamvuson@gmail.com and
hbquoc@fit.hcmus.edu.vn

Abstract. Our system (Team: HCMUS) combined rule-based and machine learning methods. The first step in which the test files were normalized and pre-processed. The pre-processing was related to the problems as: the special characters (dot in the case of abbreviation, ?, etc.), replacing the names and the dates in the brackets ([]). The document was split into the sections and paragraphs. Then the NLP tools were used for sentence splitting, POS tagging and parsing. The set of rules based on the dependence graph which were used to recognize events. In order to recognize the concepts (the 8th attribute), the UMLS and MetaMap were used. For the 9th attribute, the machine learning method was based on the features such as: document types, section types, temporal expressions (ago, today, etc.), explicit dates in the sentences and verb POS tags. For task 2a, this system achieved an overall accuracy of 0.827, F1-score of 0.389, precision of 0.367 and recall of 0.415. For task 2b, the system performed with an F1-core, precision and recall of 0.420, 0.378 and 0.472 respectively, in the strict mode and 0.648, 0.583 and 0.729 respectively, in the relaxed mode.

Keywords: Clinical Information Extraction, Clinical Relation Extraction, Natural Language Processing.

1 Introduction

ShARe/CLEFeHealth 2013 Lab offered the shared tasks: identification and normalization of disorders and normalization of abbreviations and acronyms in the clinical reports with respect to the terminology standards in the healthcare as well as the information retrieval to address the questions that the patients may have while reading clinical reports [1]. This year, ShARe/CLEFeHealth 2014 Lab has offered the three shared tasks: information visualization (task 1), information extraction (task 2) and information retrieval (task 3) [3]. We participated in dealing with task 2 in the ShARe/CLEFeHealth 2014. Task 2 is an extension of Task 1 which was done in 2013 by focusing on Disease/Disorder Template Filling. In this task, participants were provided an empty template for each disease/disorder mention; each template consisted of mention's Unified Medical Language System concept unique identifiers (CUI), mention boundaries and unfilled attribute: value slots. Participants were asked to de-

velop attribute classifiers that predict the value for each attribute: value slot for the provided disease/disorder mention. Disease/Disorder (DD) Templates consist of 10 different attributes: Negation Indicator, Subject Class, Uncertainty Indicator, Course Class, Severity Class, Conditional Class, Generic Class, Body Location, DocTime Class, and Temporal Expression[1].

In this paper, we present our approach for Task 2a and 2b of the ShARe/CLEFeHealth 2014. Our system (Team: HCMUS) consists of a machine learning based approach for the 9th attribute (DocTime Class) and a rule-based approach for the nine other attributes.

2 Methods

2.1 Pre-Processing

- ***Processing Document:*** Some punctuations (?, -, etc.) are important triggers for determining an attribute value. For example: If "-" and "?" stand in front of a disease/disorder, they determine the first attribute value is "yes" and the third attribute value is "yes" for the disease/disorder. In the dependence graph, these punctuations did not appear, so we had to replace them with the other punctuations that suit our system. Because medical data are sensitive and private, such as: all names of patients, doctors and hospitals, .etc., the data are encoded and marked by some special characters, for example: "She was transferred to [**Hospital1 27**] per recommendation of her GI specialist Dr. [**First Name (STitle) 5060**]". In addition, there are date time phrases which have been marked with special characters by annotators, for instance: "She was discharged home on [**2011-02-02**]". All special characters (like: [,*) and encoded names (like: "First Name (Stile) 5060", "Hospital1 27", etc.) will lead to incorrect parsing. Therefore, we cleaned the data by replacing encoded names with pseudo names and deleting all special characters. For example, we replaced the encoded name phrase "[**First Name (Stile) 5060]" with "Peter".
- ***Section Splitter***: Clinical notes can be considered as semi-structured data which are split into distinct sections like CHIEF COMPLAINT, HISTORY OF PRESENT ILLNESS, PAST MEDICAL HISTORY, PHYSICAL EXAMINATION, etc. Each section tends to describe events of a particular timeframe. For example, 'HISTORY OF PRESENT ILLNESS' predominantly describes events occurring before DOCTIME, whereas 'MEDICATIONS' provides a snapshot at DOCTIME and 'ONGOING CARE ORDERS' discusses events which have not yet occurred [6]. Some statistical analyses on the corpus show that 94% of Diseases/Disorders in the section "PHYSICAL EXAMINATION" are OVERLAP, 90% of Disease/Disorder in the section "CHIEF COMPLAINT" are BEFORE_OVERLAPS and 100% of Diseases/Disorders in the section "YOU SHOULD CONTACT YOUR MD IF YOU EXPERIENCE" are AFTER. The problem is how to split

[1] http://clefehealth2014.dcu.ie/task-2

clinical notes into sections. To solve this, we built a list of section names which is used to split the content into sections. The list was built semi-automatically. By experiment, we noted that sections end with colons and separate by two successive control characters \n\n. We applied regular expressions to extract a list of candidate section names. We calculated the frequency of the candidate section names. Based on this, we determined which were correct names and removed incorrect ones. However, there were some cases in which a section was presented with different names, e.g. the section DISCHARGE CONDITION appears under the names "discharge on condition" and "discharge condition". To identify these cases, we used Minimum Edit Distance to measure difference of names. Low-difference gave us a hint to check if they were variants.

- ***Paragraph Splitter*:** Each section was divided into paragraphs which were separated from each other by two successive control characters \n\n. We noted that the temporal information of a Disease/Disorder is not only in the sentence that contains it but also at the beginning of the paragraph. Therefore, we decided to split each section into paragraphs. For instance, in the following paragraph, the disease "Scaring" has temporal information "2020-05-31" which is located at the beginning of the paragraph.

  CXR ([**2020-05-31**])
  IMPRESSION: ***Scarring*** versus atelectasis in right lung base. No acute process.

- ***Sentence Splitter, POS tag and Parser:*** In this stage, natural language processing (NLP) is applied. It includes splitting sentences, tagging parts of speech, and deep parsing sentences.

2.2 Rule-based approach

This approach was applied for 9 attributes (1-10, except for 9), we used the output of the pre-processing step in order to extract the trigger sets and rule sets from the training data. The trigger sets are cue slot values corresponding to attributes of disease/disorder. Each attribute of disease/disorder has a specific trigger list. Then we enriched the trigger list by given resources. Particularly, we added triggers that have negative meaning from NegEx to the 1st attribute's trigger list. The set of rules is built manually based on linguistic information and dependency graph. The Stanford typed dependencies representation was designed to provide a simple description of the grammatical relationships in a sentence that can be easily understood and effectively used by the people who do not have linguistic expertise want to extract textual relations [2]. Fig.1. gives the representation of the dependency graph for an example sentence "*Mitral stenosis is not present and definite mitral regurgitation is not seen."* where {*not present, not seen*} are *triggers* and {*Mitral stenosis, mitral regurgitation*} are *diseases*/*disorders* in the sentence.

Next step, we built the rules based on the representation of dependency graph with the aim to identify the relation between a *disease*/*disorder* and a *trigger*. In that system, there are two rule types used: 1) Type 1 rule is a disease/disorder which is directly relevant to a trigger and 2) Type 2 rule is a disease/disorder which is indirect-

ly relevant to a trigger through another disease/disorder. Each attribute has its own trigger list. The rules are performed as follows:

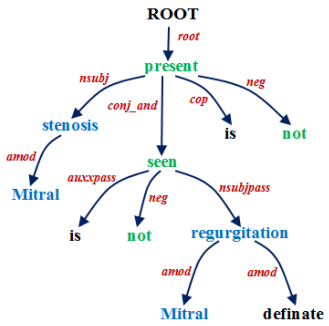

Fig. 1. Representation of the dependency graph

Type 1 rule:

({*relation* = rel_label}{*governor* = DD}{*dependent* = trigger ∈ trigger_list}) → (DD: norm_value)

Where:

re_label *is a relation label between the disease/disorder and the trigger on the dependency graph*

DD *is a candidate disease/disorder*

trigger_list *is trigger list of attribute*

norm_value *is a norm slot value of attribute*

Type 2 rule:

({*relation* = rel_label}{*governor* = DD1 ∈ DD1_list}{*dependent* = DD2 }) → (DD2: norm_value1)

Where:

DD1 *has a norm_value1 which was identified by type 1 rule*

DD1_list *is list of DD1*

DD2 *is a candidate disease/disorder*

For example, the rule set identifies 1st attribute values in Fig.2. as follows:

Type 1 rule:

({*relation* = "neg"}{*governor* = "clubbing"}{*dependent* = "No"}) → ("clubbing": Yes)

Type 2 rule:

({*relation* = "conj_or"}{*governor* = "clubbing"}{*dependent* = "cyanosis"}) → ("cyanosis": Yes)

Or

({*relation* = "conj_or"}{*governor* = "clubbing"}{*dependent* = "edema"}) → ("edema": Yes)

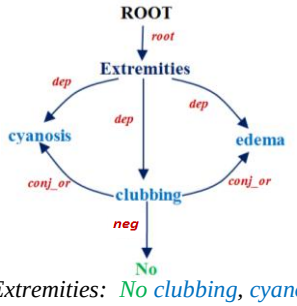

Fig. 2. An example "*Extremities: No clubbing, cyanosis, or edema.*"

2.3 Machine learning-based approach

For the 9th attribute, we used the machine learning approach, SVM. The document was transformed to feature vectors for the machine learning algorithm. Features were extracted as follows:

- ***Document type feature:*** document type that the Disease/Disorder appears.
- ***Section feature:*** as described in the pre-processing phase, a section is a feature to identify relation between Disease/Disorder and DocTime.
- **Temporal expression**: There are temporal expressions that help to predict the output, such as "ago", "today", "at present", etc.
- ***Explicit date feature:*** Explicit dates are date strings that are explicitly annotated in the clinical notes by the annotators. They can be identified by using regular expression. However, it is necessary to determine the scope of the explicit date. We divided the explicit date into two levels: 1) Sentence scope and 2) Paragraph scope. For example: disease/disorder "headache" in the sentence "In [**2015-01-14**], the patient had headache …" has explicit date (2015-01-14) in Sentence scope. An explicit date having the Paragraph scope means that this explicit date may link to all disease/disorder within the paragraph. In the experiments, we found the explicit dates in the Paragraph scope which means usually occur in the PERTINENT RESULTS section. The relation between this explicit date and the admission date, the discharge date is used as a feature for our classifier.

Verb POS tags: based on the parser, we identify verbs which link to the Disease/Disorder and their POS tags.

3 Resources

Our system used the resources such as Stanford NLP tool[2], NegEx project[3], MetaMap tool[4], LibSVM[5], Weka tool[6] and UMLS[7]. We used the Stanford NLP tools for

[2] http://nlp.stanford.edu/index.shtml
[3] https://code.google.com/p/negex/
[4] http://metamap.nlm.nih.gov/

pre-processing texts, WEKA for classifying the 9th attribute, and MetaMap for identifying candidate UMLS concepts. MetaMap creates its final UMLS concept mapping by choosing appropriate candidates that cover as much of the input text as possible.

4 Results

For the attributes from 1 to 7, the rule-based approach performs well. The rules based on the dependence graph have contributed to the high performance. The highest accuracy among the predicted attributes is 0.995 for the subject class (Table 1). The machine learning used to predict the 9th attribute did not produce the high performance. The reason may be due to the feature set which is not good enough to recognize the document time.

Table 1. Predict each attribute's normalization slot value (Task 2a)

	Attributes	Accuracy	F1-score	Precision	Recall
1	Negation Indicator (NI)	0.910	0.803	0.735	0.885
2	Subject Class (SC)	0.995	0.736	0.760	0.713
3	Uncertainty Indicator (UI)	0.877	0.385	0.274	0.646
4	Course Class (CC)	0.937	0.410	0.317	0.577
5	Severity Class (SV)	0.961	0.662	0.626	0.702
6	Conditional Class (CO)	0.899	0.441	0.340	0.625
7	Generic Class (GC)	1.000	0.000	0.000	0.000
8	Body Location (BL)	0.551	0.330	0.309	0.354
9	DocTime Class (DT)	0.306	0.306	0.306	0.306
10	Temporal Expression (TE)	0.830	0.313	0.337	0.292
	Average	0.827	0.389	0.367	0.415

For Task 2b, our system achieves an F1-score of 0.420 in the strict evaluation (Table 2) and F1-score of 0.648 in the relaxed evaluation (Table 3) respectively, which suggests a rule-based approach that can not identify an exact cue span representing an attribute value.

5 http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/
6 http://www.cs.waikato.ac.nz/ml/weka/
7 http://www.nlm.nih.gov/research/umls/

Table 2. Task 2b: Predict each attribute's cue slot value (strict)

	Attributes	F1-score	Precision	Recall
1	Negation Indicator (NI)	0.622	0.559	0.699
2	Subject Class (SC)	0.384	0.397	0.372
3	Uncertainty Indicator (UI)	0.207	0.147	0.346
4	Course Class (CC)	0.388	0.301	0.545
5	Severity Class (SV)	0.686	0.649	0.726
6	Conditional Class (CO)	0.248	0.192	0.352
7	Generic Class (GC)	0.000	0.000	0.000
8	Body Location (BL)	0.419	0.392	0.451
9	DocTime Class (DT)	-	-	-
10	Temporal Expression (TE)	0.260	0.281	0.242
	Average	**0.420**	**0.378**	**0.472**

Table 3. Task 2b: Predict each attribute's cue slot value (relaxed)

	Attributes	F1-score	Precision	Recall
1	Negation Indicator (NI)	0.817	0.735	0.919
2	Subject Class (SC)	0.936	0.967	0.907
3	Uncertainty Indicator (UI)	0.386	0.275	0.646
4	Course Class (CC)	0.447	0.348	0.628
5	Severity Class (SV)	0.710	0.672	0.752
6	Conditional Class (CO)	0.441	0.340	0.625
7	Generic Class (GC)	0.000	0.000	0.000
8	Body Location (BL)	0.750	0.701	0.807
9	DocTime Class (DT)	-	-	-
10	Temporal Expression (TE)	0.354	0.383	0.329
	Average	**0.648**	**0.583**	**0.729**

5 Conclusion

We applied the rule-based approach for all attributes, except for the 9th attribute which was processed by the machine learning approach. In Task 2a, our system achieved an overall accuracy of 0.827, F1-score of 0.389, precision of 0.367 and recall of 0.415. In Task 2b, our system performed with an F1-core, precision and recall of 0.420, 0.378 and 0.472 respectively, in the strict mode and 0.648, 0.583 and 0.729, respectively, in the relaxed mode. Further improvements will be likely to be feasible by adding new features to the machine learning model and normalization of rule set, as well as adding an approach that combines the rule-based and machine learning for all attributes. These tasks will be the focus of interest in future work.

References

1. Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R. South, Danielle L. Mowery, Gareth J. F. Jones, Johannes Leveling, Liadh Kelly, Lorraine Goeuriot, David Martinez, and Guido Zuccon.: Overview of the ShARe/CLEF eHealth Evaluation Lab 2013.
2. Marie-Catherine de Marneffe and Christopher D. Manning. Stanford typed dependencies manual. September 2008.
3. 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.
4. N Elhadad, W Chapman, T O'Gorman, M Palmer, G Savova. The ShARe Schema for the Syntactic and Semantic Annotation of Clinical Texts. In preparation.
5. Strötgen J, Gertz M. HeidelTime. High quality rule-based extraction and normalization of temporal expressions. Proceedings of the 5th International Workshop on Semantic Evaluation. Los Angeles, California: Association for Computational Linguistics. 2010.
6. William F. Styler, et al. Temporal Annotation in the Clinical Domain. Transactions of the Association for Computational Linguistic. 2014.