International Journal of Bio-Science and Bio-Technology
Vol.8, No.4 (2016), pp. 111-124
http://dx.doi.org/10.14257/ijbsbt.2016.8.4.13

Revisiting Medical Entity Recognition through the Guidelines of the Aurora Initiative

Praveen Kumar[1], Sabah Mohammed[2], Arnold Kim[3] and Jinan Fiaidhi[4]

[1, 2, 3, 4] *Computer Science, Lakehead University, Thunder Bay, Canada*
[1]*pkumar2@lakeheadu.ca* , [2]*sabah.mohammed@lakeheadu.ca,*
[3]*afkim@lakeheadu.ca,* [4]*jfiaidhi@lakeheadu.ca*

Abstract

Clinical Document Processing is growing importance because of unstructured nature of clinical notes as well as limitation of crucial time of clinical professionals to analyses the unstructured clinical notes. Named entity recognition (NER) is a subtask of Clinical documentation processing which is important not only for text analysis but knowledge extraction. Although there are a number of clinical named entity recognition systems, they lack user flexibility and NER scalability. Clinical NER is a challenging work which required consistent research to improve clinical documentation.

Accordingly, in this paper, keeping an eye on user's flexibility, we combined the NER technique with DSL (Domain Specific Language) based user queries. This research focused to produce a prototype system which allows the user to input their queries about a clinical text in a syntax free language which will be reformulate into DSL format in background. The reformulated query then matches against the rules defined by using the DSL to get the matched rule-type. The DSL is created using Xtext framework specifically to create NER rules easily. Then NER is done as per the found NER rule-types. We used the lingpipe API to do the NER using unsupervised technique (dictionary based approach). Again considering user flexibility, research also focused on graphical visualization of the annotated recognized entities, flexibility to store the annotated document into database for later use as well as can conversion the recognized entities into CDA (Clinical Document Architecture) format for interoperability. This research is initiated and inspired by the Aurora research initiative which is an ongoing attempt lead by Dr. Arnold Kim to integrate the design of clinical documentation workflows from the physician perspective that starts with variety DSLs and ends with series of interpretations and analytics in the background

Keywords*: Electronic medical records, Text Analysis, Named Entity Recognition (NER), Domain Specific Language (DSL)*

1. Introduction

Proper and accurate clinical documentation is always being important for healthcare industry, but in today's shifting healthcare environment, it has become even more crucial than perhaps ever before. Documentation is critical for patient care, not only because it validates the care that was provided, shares key data with subsequent caregivers and optimizes claims processing but also for text mining, text analysis as well as interoperability. As such, clinical documentation improvement (CDI) programs are important to any facility that recognizes the necessity of complete and accurate patient documentation.

At the center of the problem is that physicians are extremely busy and because of that, they do not link main points on clinical documentation. One way is to train physicians, and give them time to see the relevance in improving their documenting. On the other hand, use of latest technology to do automatic text analysis is equally important. It helps in understanding the clinical texts, mine knowledge from them and create better decision systems Automatic conversion of clinical text to CDA format is another very important for interoperability of the clinical documents between care givers and document analyzers who uses different document clinical documentation standards. Named Entity Recognition is one of the important pre-processing steps of text analysis which focused on recognizing interested named entities in the texts via using technological advances. Clinical Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) seeks to locate and classify clinical named-entities in clinical text into pre-defined categories such as the medication, symptoms, allergies, names of physician, important dates, lab test etc.

Aurora: Is a multidisciplinary initiative focused on developing, implementing and measuring the feasibility of health care provider's capacity to express long-held schematic thought processes in the form of a formal language that adheres to traditional medical documentation idioms, in order to improve the construction and validation of patient care planning. The goals are to maximize usability, efficiency, error elimination, measurability, collaboration, & policy enforcement. These attributes will define next generation EHR's which will offer health care providers and administrators a powerful design environment for patient care planning. One of the task of aurora is to automatically produce formatted clinical documents from clinical professional syntax based notes using DSL approach.

Our research is inspired by the Aurora initiative in using DSL but instead of creating document from clinician's syntax based notes, our clinical document parser is guided by a DSL in the form of clinician questions which user can input in syntax free language (English) to identify entities at the clinical documents (e.g. medications, allergies, symptoms, etc.).

2. Literature Review

There are many different techniques used by researchers to recognise the named entities in unstructured texts namely supervised, unsupervised and semi-supervised.

Supervised NER techniques needs complete and reliable training dataset. The training data is feed into Machine Learning algorithms to learn and work itself without any further need of human interventions. Few of the supervised methods are SVM (Support Vector Machine), HMM (Hidden Markov Model), ME (Maximum Entropy), CRF (Condition Random Fields) etc. Lots of research is done in Bio-Medical NER domain. Kanya[5] *et al.* tried many techniques including BLS, HMM, TBL, SVM and their combinations to do Biomedical NER. Li[2] *et al.* used Semi-CRF method which is an extension of CRF method and produced better results. Ju[3] *et al.* and Betina[6] *et al.* also used SVM for Biomedical NER. Liao[4] *et al.* used Skip-Chain CRF method especially to solve long range dependency problem in Biomedical NER. Vijayraghavan[1] *et al.* used SVM techniques to specifically recognise anatomical phrases in Clinical documents. They first classify all words in five classes a) Begin [B], b) End [E], c) Inside [I], d) Single [S], and e) Outside [O]. To have a good training dataset they performed boundary detection, tokenisation, POS and semantic tagging, Word-sense disambiguation and then used high-recall sentence level phrase parser and then filtered by domain experts. Then we used this highly verified training data to do NER on test dataset using SVM.

Unsupervised NER techniques needs dictionaries or pre-defined source of information look-ups and work on direct string matching. The drawback is massive dictionaries for each class which needed periodic updates. Supervised model results are more promising then unsupervised model but required complete and reliable training data. Ling[10] *et al.* used MetaMap (symptoms) and MedEx (medication) dictionaries to do NER in clinical domains. He also used Standford CoreNLP tool to segment the document and NegEx to negative sections like allergies or family history. Niazi[11] *et al.* adopted a novel unsupervised NER approach after looking at the saturation of supervised techniques. They automated UMLS concepts as signature vectors and used for NER in Bio-Medical texts along with other method noun-phrase chucking, TF, IDF and cosine similarity.

Rule based NER techniques needs heuristic rules to recognise patterns (named entities) in the text. This technique need lots of expertise to design rules for detecting particular named entity. Chen[7] *et al.* took input from user as a keyword. Then look for its sematic nature and then fire rules to recognise and annotate entity at runtime. Faitha[8] *et al.* divided the problem in phases. They first segmented the documented by using rules [pattern matching] and then did tokenizing & tagging in each section and finally used rules again to recognise named entities in each section. Champion [9] *et al.* solved problem in cTAKES NER system by extending SystemT to work for huge dataset.

Semi-supervised NER techniques are a combination of supervised and unsupervised techniques where researchers trying to have the benefits of both the techniques. In Bio-medical domain, Gu [12] *et al.* combined dictionary look-up and SVM whereas Wei [17] *et al.* combined pattern matching (rules) and CRF. In clinical domain, Gong [13] *et al.* combined rules and dictionaries look-up techniques, Apostolova[14] *et al.* combined rules and SVM, Han[15] *et al.* combined dictionaries with enhanced SVM called EK-SVM-KNN (Extended SVM-KNN) whereas Feng[16] and Wei[17] *et al.* combined rules and CRF combined rules and CRF techniques.

Dehgam [18] *et al.* presented a deep insight of named entity recognition focusing clinical documents and existing challenges which needs to be focused. Groza [19] *et al.* compared existed NER tools cTAKES, NCBO Annotator, BeCAS and Metamap.

Apart from NER research problem in clinical text, there is one another challenge of converting the clinical documents to interoperable format which can be easily moved across or outside healthcare organisation for better patient care as well as further research and educational purposes. CDA is a globally accepted clinical document standard from HL7 which is very flexible to store and maintain the semantic annotation information within itself for later reuse either for improved patient care, for external research looking for data mining and also for information exchange or interoperability between different hospitals.

Treins [20] *et al.*, DuVall [22] *et al.* discussed the importance of HL7 CDA documents to retain the annotation of structural and sematic concepts in medical documents for interoperability. DuVall [22] *et al.* also focused towards storing the annotated CDA document in a corpus for further researches. Huang [21] *et al.*, Lin [23] focused on generation of structured documents. Huang [21] *et al.*, developed a model to generate standardised CDA R2 document from EMR (Excel file) whereas Lin [23] proposed a pipeline to generate CDA entries from free-text. Lin [23] *et al.*, used the clinical named entity recognition and annotation tool cTAKES results which is based on UIMA-Common Analysis System with XML representation to generate CDA XML documents.

I found two important problems which can be improved:

First, Improving the processing of clinical documentation by recognising named entity in the clinical documents from user's perspective by using DSL (Domain Specific language). DSL will be used to form rules for particular named entity recognition. User should have allowed to input queries in syntax free language.

Second, converting annotated recognized entities into CDA format for making them interoperable so that this work not only facilitate better patient care but improve education, training and future research across the healthcare organizations.

4. Existing Methods and Tools

Researchers are continuously putting efforts in developing tools to do named entity recognition in clinical documents by using different techniques and approaches. Below are the some of the major tools developed by the researcher to do the CDI (clinical documentation improvements).

Apache UIMA (Unstructured Information Management Application) is a project targeting to support a thriving community of users and developers of UIMA frameworks, tools, and annotators, facilitating the analysis of unstructured content such as text, audio and video. UIM applications analyse large volumes of unstructured information in order to discover knowledge that is relevant to the end user. UIM applications accepts a plain text and identify entities, such as persons, places, organisations or relations such as work-for or located-at. UIMA enables applications to be decomposed into components, for example "language identification" => "language specific segmentation" => "sentence boundary detection" => "entity detection (person/place names etc.)". Each component implements interfaces defined by the framework and provides self describing metadata via XML descriptor files. The framework manages these components and data flow between them. Components are written in Java or C++ and the data that flows between components is designed for efficient mapping between these languages. UIMA additionally provides capabilities to wrap components as network services, and can scale to very large volumes by replicating processing pipelines over a cluster of networked nodes.

cTAKES is another important tool developed by Mayo Clinic to do NER in clinical documents. It targets to recognise four entities disorder (diseases), sign/symptoms, procedures and drugs. It uses a hybrid approach to recognise entities. It uses entities dictionaries along with terms maintained by the mayo clinic. It targeted to find the entities for non-lexical variations by doing the permutation of the head and modifier within the noun phrases. It also targeted identifying multiple terms in the same span. It used pattern base approach of Negation annotator (Negex Algorithm) for findings words and phrases negative near named entities. It acts like a chain of operation where it first finds the start and end of texts, then it finds the terminology code and concept unique identifier (cui), then find the negated entities by using the status associated with the named entities like allergy, family history etc. and finally visualises the found entities. It faced challenges in recognising complex level of synonymy, word sense disambiguation and coordination structure interpretation.

MedEx tool process free-text clinical records to recognize medication names and signature information, such as drug dose, frequency, route, and duration. It uses a context-free grammar and regular expression parsing to process free text clinical notes. After finding medication information, it maps to RxNorm and UMLS concepts at the most specific match it can find (e.g., medication name + strength would be preferred to medication name alone). MedEx is a medication parser developed by using semantic types and patterns in a much finer granularity. First of all, as a pre-processing step MedEx detects existing sentence boundary by using a rule-based program. Then MedEx does semantic tagging of each input sentence into token and label proper words with a semantic category. Then it disambiguates the ambiguous tags by using context based rules. It combines a lookup tagger and a regular expression tagger to tag different semantic pieces of a medication. It uses a lexicon files of drug names from RxNorm by combining terms from normalized drug forms including IN (Ingredient, eg, Fluoxetine), BN (Brand name, eg, Prozac), SCDC (Ingredient+Strength, eg, Fluoxetine 4 mg/ml), SCDF (Ingredient+Form, eg, Fluoxetine Oral solution), and SCD (Ingredient+Strength+Form, eg, Fluoxetine 4 mg/ml Oral solution). If a drug finding is tagged as SCDC, SCDF, or SCD, it is straightforward to further decompose it into DrugName, Strength, and Form, based on relations within the RxNorm after removal of ambiguous English words manually reviewed by a physician and adding full form of some abbreviation into the lexicon file. They combined two taggers in a sequential manner (the lookup tagger followed by the regular expression tagger). The lookup tagger maps a drug name to its longest match in the lexicon file. Other types of information are more suitable for a regular expression tagger. For example, frequency information such as 'q4h or q6h' can be easily captured by defining regular expressions such as 'q\dh'. The parsing component of MedEx uses a context-free grammar to parse textual sentences into structured forms, via a Chart Parser, a dynamic programming parsing method. If the Chart parser fails, a regular expression based Chunker in Natural Language Tool Kit is used to process the medication sentences. For example, medication phrases can be defined as regular expressions such as 'DrugName (DOSE|FORM|RUT|FREQ)*', which indicates a medication phrase can be composed by one drug name followed by zero or more signature items including 'Dose', 'Form', 'Route', and 'Frequency'

MedLEE (Medical Language Extraction and Encoding System) is developed by scientist Carol Friedman as a general natural-language processor that identifies clinical information in narrative reports and maps that information into a structured representation containing clinical terms. It was originally designed for decision support applications in the domain of radiology reports of the chest, and was extended to other domains, such as mammography and discharge summaries later. The natural-language processor provides three phases of processing, all of which are driven by different knowledge sources. The first phase performs the parsing. It identifies the structure of the text through use of a grammar that defines semantic patterns and a target form. The second phase, regularization, standardizes the terms in the initial target structure via a compositional mapping of multi-word phrases. The third phase, encoding, maps the terms to a controlled vocabulary. Radiology is the test domain for the processor and the target structure is a formal model for representing clinical information in that domain.

MetaMap is a highly configurable program developed by Dr. Alan (Lan) Aronson at the National Library of Medicine (NLM) to map biomedical text to the UMLS Metathesaurus or, equivalently, to discover Metathesaurus concepts referred to in text. MetaMap uses a knowledge-intensive approach based on symbolic, natural-language processing (NLP) and computational-linguistic techniques. Besides being applied for both IR and data-mining applications, MetaMap is one of the foundations of NLM's Medical Text Indexer (MTI) which is being used for both semiautomatic and fully automatic indexing of biomedical literature at NLM.

KnowledgeMap (KM) is a NLP system developed at Vanderbilt University and it has been used to extract medical concepts from clinical and education documents. The KM concept identifier uses lexical resources partially derived from the UMLS (SPECIALIST lexicon and Metathesaurus), heuristic language processing techniques, and an empirical scoring algorithm. KM differentiates among potentially matching Metathesaurus concepts within a source document.

HITEx (Health Information Text Extraction) is an open-source natural language processing (NLP) software application developed by a group of researchers at the Brigham and Women's Hospital and Harvard Medical School. HITEx is built on top of Gate framework and uses Gate as a platform. HITEx consists of the collection of Gate plug-ins that were developed to solve problems in medical domain, such as principal diagnoses extraction, discharge medications extraction, smoking status extraction and others. HITEx works by assembling these plug-ins into pipeline applications, along with other standard NLP plug-ins (some of which are part of Gate, such as Part-of-Speech tagger or Noun Phrase Chunker). Each plug-in in a pipeline may use the output of the previous plug-in. Power users are given full control over the plug-in parameters and the order of plug-ins in the application. General users may benefit from pre-configured pipeline applications that solve common medical problems, such as principal diagnoses extraction, discharge medications extraction, smoking status extraction and others.

5. The Prototype Design

This section describes the design of the new NER system developed by combining it with the DSL.

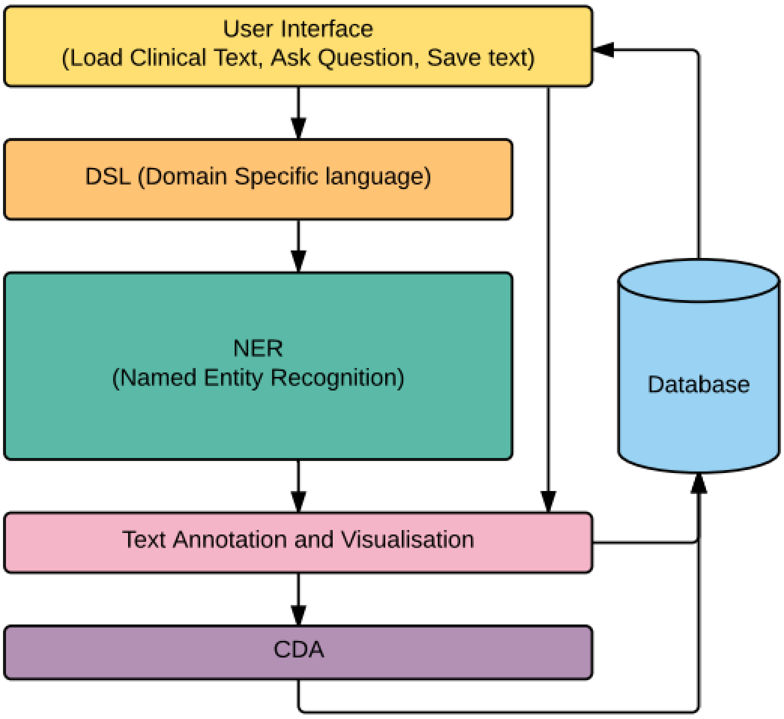

User Interface
(Load Clinical Text, Ask Question, Save text)
DSL (Domain Specific language)
NER
(Named Entity Recognition)
Database
Text Annotation and Visualisation
CDA

Figure 1. The Prototype Architecture

The Figure 1 describes system architecture showing the different layers of the system and how they will be connected to each other. The User Interface will be the top layer which allows the user to load and save clinical texts as well as to ask question about the details in them. The DSL component just below which accepts the user question is a syntax free language and transform it into syntax language to match it with the defined DSL NER rule-types. If matched rule is found, application will do NER in NER layer beneath DSL layer. NER module will use the matched DSL rule-type to find the region of interest and the result will be passed to the Text annotation and Visualization layer to annotate the clinical document and visualize it to the user in an elegant manner. At the end results will be stored in the database and gives option to users if he wants the convert the recognized named entities into CDA (Clinical Document Architecture) format for interoperability.

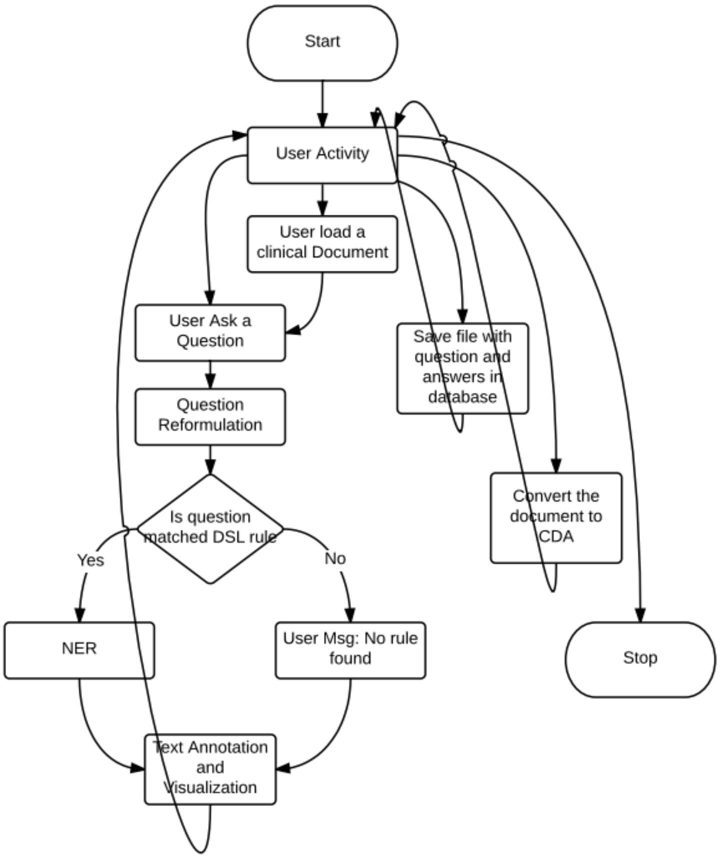
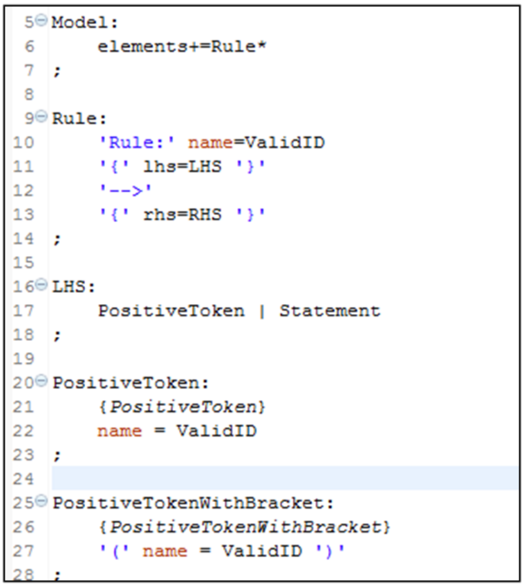

Start
User Activity
User load a clinical Document
User Ask a Question
Question Reformulation
Is question matched DSL rule
Yes
No
NER
User Msg: No rule found
Text Annotation and Visualization
Save file with question and answers in database
Convert the document to CDA
Stop

Figure 2. System Processing

The Figure 2 describes the the work flow of the proposed method to recognize the name entities in clinical documents. User can load a clinical document and ask a question about it in a syntax free language. The question will be reformulated and tried to find the matched DSL NER rule-type. If no matched rule-type found, proper message will be displayed to user otherwise NER is performed for the recognized rule type. The recognized entities then annotated with colorful background and visualized to the user. User have other optional activities like storing annotated document along with other information into the database, view the recognized entities into graphical view and converting the recognized entities into CDA format. User can exit anytime the application prototype by using exit button.

6. Implementation Details

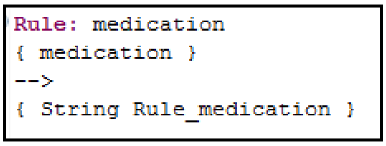
We designed a novel prototype of NER system. We created a Domain Specific Language (DSL) using Xtext Framework to form NER rules. Figure 3 shows the snippet of the defined grammar for DSL.



```
Model:
    elements+=Rule*
;

Rule:
    'Rule:' name=ValidID
    '{' lhs=LHS '}'
    '-->'
    '{' rhs=RHS '}'
;

LHS:
    PositiveToken | Statement
;

PositiveToken:
    {PositiveToken}
    name = ValidID
;

PositiveTokenWithBracket:
    {PositiveTokenWithBracket}
    '(' name = ValidID ')'
;
```

Figure 3. DSL Grammar Snippet

The DSL allows to form rules in a very easy and flexible manner. Figure 4 shows the example of rule medication as:



```
Rule: medication
{ medication }
-->
{ String Rule_medication }
```

Figure 4. DSL Grammar Rules

User is allowed to input his query in natural language (English) which then reformulate application into defined DSL format. I used Stanford NLP parser and tagger to parse the user question along with WordNet Library to get the synonyms of the word in user question in question reformulation. Then reformulated question is matched against defined NER rule-types. Figure 5 shows the pseudo code for reformulation of user query and finding the matched NER type.

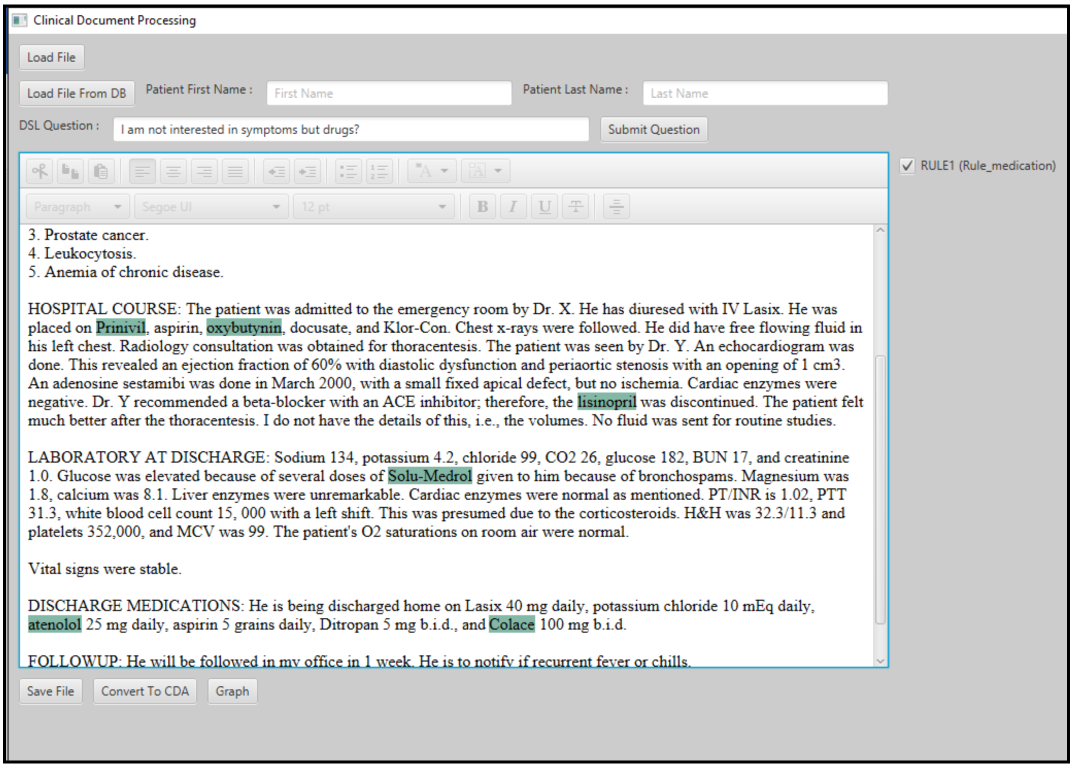
Table 1. PusedoCode

```
__________________________________________________
Algorithm 1 Finding NER Rules
__________________________________________________
processQuestion(q)
Input : User question q in natural language, where q has medium complexity
Output : List of NER Rule-Types
if(q is not empty) then
  words = Filter(q) //Nouns,Coordinating conjunction,Adverb
  words = Reform(words) //Validate & conversion to Logical operator and adjustment
  if(words.length <= 3) then
          if(words.length==1) then
                    //To find Direct Rules
                    l:list[String] = getSynonyms(words[0])
                    matchRuleList:List[] //emptylist
                    foreach(i in l) {
                      queryDSLFormats = reformQuery(i)
                      r = MatchDSLRule(queryDSLFormats)
                      if(r is Valid) then
                        add r in matchRuleList
```

```
             end
            }
            removeDuplicateRules(matchRuleList)
            return matchRuleList
      else if(words.length==3) then
            //To find Direct Rules
            l1:list[words] = words[0] find all synonyms
            l2:list[words] = words[2] find all synonyms //word[1] is operator
            matchRuleList:List[] //emptylist
            foreach(i in l1) {
             foreach(j in l2) {
              queryDSLFormats = reformQuery(i,j);
              queryDSLFormats = reArrangeQuery(queryDSLFormats);
              r = MatchDSLRule(queryDSLFormats)
              if(r is Valid) then
               add r in matchRuleList
              end
             }
             removeDuplicateRules(matchRuleList)
            }
            if(matchRuleList is not empty) then
             return matchRuleList
            else
             //Find compound Rules
             l:list[String] = getSynonyms(words[0])
             mRL1:List[] //emptylist
             foreach(i in l) {
              queryDSLFormats = reformQuery(i)
              r = MatchDSLRule(queryDSLFormats)
              if(r is Valid) then
               add r in mRL1
              end
             }
             removeDuplicateRules(mRL1)
             l:list[String] = getSynonyms(words[2])
             mRL2:List[] //emptylist
             foreach(i in l) {
              queryDSLFormats = reformQuery(i)
              r = MatchDSLRule(queryDSLFormats)
              if(r is Valid) then
               add r in mRL2
              end
             }
             removeDuplicateRules(mRL2)
             foreach(i in mRL1) {
              foreach(j in mRL2) {
               queryDSLFormats = reformQuery(i,j);
               queryDSLFormats = reArrangeQuery(queryDSLFormats);
               r = MatchDSLRule(queryDSLFormats)
               if(r is Valid) then
                add r in matchRuleList
               end
              }
              removeDuplicateRules(matchRuleList)
             }
             return matchRuleList
            end
      end
     else
      //complex query
      return (empty list)
     end
else
     return (empty list)
end
```

For example, when user input a query in generic form say *"I am not interested in symptoms but drugs?"* then user query will be reformulated into DSL format after dropping of all the syntactic sugar and the query will take the as *"~symptoms & drugs"*. Then for each of the synonyms of found nouns "symptoms" and "drugs", a new query is generated and matched against the defined rule. If a matched found rule found it stops and send the matched rule to NER module. If no matched rule found, then for each noun "~symptoms" and "drugs" and for all their synonyms matched separately against all the defined rule to find the matched rules. Then it looks for combined rule by combining the find rules using same logical operator if any. If found it return the combined rule otherwise it returned the separately found rule. Here the user query matched the define rule "*Rule: medication*" and return with string "*Rule_medication*".

The matched NER rule type is passed on to the NER routine. Then if NER module supports NER for found rule type, it will be done in background by using either rule based approach or unsupervised technique (dictionary) which bypasses the human intervention however dictionaries needs to have the updated dataset of entities. We used lingpipe API for NER. The result is then visualized to the user in form of annotated recognized entities in the clinical documents using colored background. Figure 5 shows the annotated recognized entities.



Patient First Name : First Name
Patient Last Name : Last Name

DSL Question : I am not interested in symptoms but drugs?

☑ RULE1 (Rule_medication)

3. Prostate cancer.
4. Leukocytosis.
5. Anemia of chronic disease.

HOSPITAL COURSE: The patient was admitted to the emergency room by Dr. X. He has diuresed with IV Lasix. He was placed on Prinivil, aspirin, oxybutynin, docusate, and Klor-Con. Chest x-rays were followed. He did have free flowing fluid in his left chest. Radiology consultation was obtained for thoracentesis. The patient was seen by Dr. Y. An echocardiogram was done. This revealed an ejection fraction of 60% with diastolic dysfunction and periaortic stenosis with an opening of 1 cm3. An adenosine sestamibi was done in March 2000, with a small fixed apical defect, but no ischemia. Cardiac enzymes were negative. Dr. Y recommended a beta-blocker with an ACE inhibitor; therefore, the lisinopril was discontinued. The patient felt much better after the thoracentesis. I do not have the details of this, i.e., the volumes. No fluid was sent for routine studies.

LABORATORY AT DISCHARGE: Sodium 134, potassium 4.2, chloride 99, CO2 26, glucose 182, BUN 17, and creatinine 1.0. Glucose was elevated because of several doses of Solu-Medrol given to him because of bronchospams. Magnesium was 1.8, calcium was 8.1. Liver enzymes were unremarkable. Cardiac enzymes were normal as mentioned. PT/INR is 1.02, PTT 31.3, white blood cell count 15, 000 with a left shift. This was presumed due to the corticosteroids. H&H was 32.3/11.3 and platelets 352,000, and MCV was 99. The patient's O2 saturations on room air were normal.

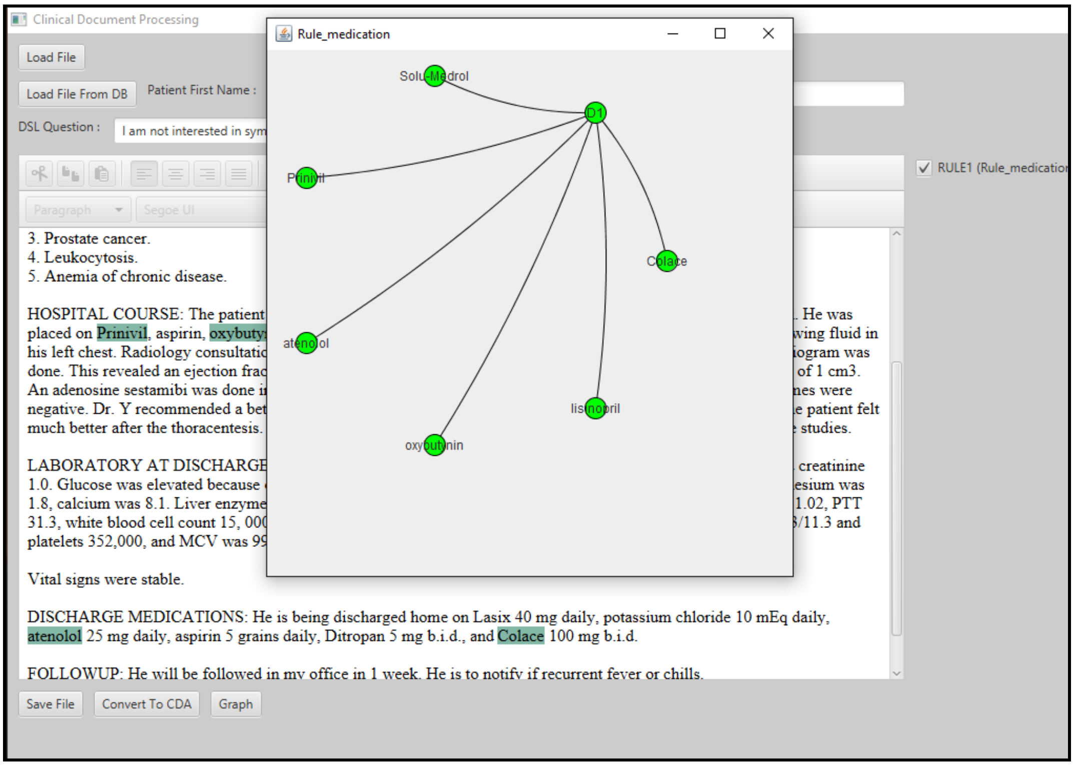
Vital signs were stable.

DISCHARGE MEDICATIONS: He is being discharged home on Lasix 40 mg daily, potassium chloride 10 mEq daily, atenolol 25 mg daily, aspirin 5 grains daily, Ditropan 5 mg b.i.d., and Colace 100 mg b.i.d.

FOLLOWUP: He will be followed in my office in 1 week. He is to notify if recurrent fever or chills.

Figure 5. Application GUI Screen

Users have the flexibility to select or deselect a particular rule-type of recognized named entities using checkbox for that NER rule to analyze them easily. User also have the flexibility to view the resulted entities in a graphical view so he doesn't need to scroll through the whole document. Figure 6, shows the graphical view of the annotated recognized entities. User can also store the results in the database which can be retrieved anytime later when needed, by using the patient name. User also have the flexibility to convert the document into CDA format for interoperability.



DSL Question : I am not interested in sym

3. Prostate cancer.
4. Leukocytosis.
5. Anemia of chronic disease.

HOSPITAL COURSE: The patient . He was
placed on Prinivil, aspirin, oxybuty wing fluid in
his left chest. Radiology consultatic iogram was
done. This revealed an ejection frac of 1 cm3.
An adenosine sestamibi was done i nes were
negative. Dr. Y recommended a bet e patient felt
much better after the thoracentesis. studies.

LABORATORY AT DISCHARGE creatinine
1.0. Glucose was elevated because esium was
1.8, calcium was 8.1. Liver enzyme 1.02, PTT
31.3, white blood cell count 15, 000 /11.3 and
platelets 352,000, and MCV was 99

Vital signs were stable.

DISCHARGE MEDICATIONS: He is being discharged home on Lasix 40 mg daily, potassium chloride 10 mEq daily, atenolol 25 mg daily, aspirin 5 grains daily, Ditropan 5 mg b.i.d., and Colace 100 mg b.i.d.

FOLLOWUP: He will be followed in my office in 1 week. He is to notify if recurrent fever or chills.

Figure 6. Graphical Visualization

7. Discussion and Conclusion

We did experiment on 10 discharge summaries. We fed them into our application and did NER on it to find the symptoms and medications in them and results are as below:

Experiment1: Recognizing "Symptoms" Named Entities, Precision: 98%, Recall: 68%, F-Score(Accuracy): 81%

Experiment2: Recognizing "Medication" Named Entities, Precision: 83%, Recall: 42%, F-Score(Accuracy): 56%

The reason for less accuracy of "medication" is the complex structure of the medications. Medication is not a single word but composed of three components which are name dose frequency. Because of complex structure it is hard to accurately recognize medication by using only dictionary model. It need extra support to get the dependencies of the nearby words.

The aim of this research is to improve clinical documentation processing. Hence we produced a prototype NER system which is more user friendly. User can ask queries in free text and the system will automatically understand the question, reformulate it to DSL format and to find out the matched NER rule type. Next the system does the NER using dictionary based approach and visualize the recognized named entities in the clinical text in a nice way. User have the flexibility to store the annotated text into database view entities in graphical view or convert recognized data into CDA format.

However, we have created a user flexible improved clinical document processing system, but there are many areas in the system which needs improvement. Due to limitation of time and resources, we were unable to enhance these areas. Here is the list of several ideas that have been considered for our future research work

- NER results for medications can be improved by using Stanford dependency parser or other similar tool

- DSL Grammar can be enhanced to create more robust rules to handle complex user question
- Conversion to CDA can be more accurate by finding the actual section of xml where we need to add the recognised entities
- Visualization can user interactive for more user friendly

References

[1] V. Bashyam, and R. K. Taira. "Identifying Anatomical Phrases in Clinical Reports by Shallow Semantic Parsing Methods." InComputational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on, **(2007)**, pp. 210-214.

[2] L. Yang, and Yanhong Zhou, "Two-phase biomedical named entity recognition based on semi-CRFs." In Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on, **(2010)**, pp. 1061-1065.

[3] Z. Ju, J. Wang, and F. Zhu, "Named entity recognition from biomedical text using SVM." In Bioinformatics and Biomedical Engineering,(iCBBE) 2011 5th International Conference on, **(2011)**, pp. 1-4.

[4] Liao, Zhihua, and H. Wu, "Biomedical Named Entity Recognition Based on Skip-Chain CRFS." In Industrial Control and Electronics Engineering (ICICEE), 2012 International Conference on, **(2012)**, pp. 1495-1498.

[5] N. Kanya, and T. Ravi, "Modelings and techniques in named entity recognition: an information extraction task", **(2012)**, pp. 104-108.

[6] A. J. Be, and G. S. Mahalakshmi, "Named entity recognition for tamil biomedical documents". In Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on, **(2014)**, pp. 1571-1577.

[7] Chen, C. Huang, X. O. Ping, Z. J. Wang, S. L. Hsieh, L. C. Chen, Y. J. Tseng, C. W. Hsu, and F. Lai, "The keyword-based and semantic-driven data matching approach for assisting structuralizing the textual clinical documents." In Biomedical Engineering and Informatics (BMEI), 2010 3rd International Conference on, vol. 6, **(2010)**, pp. 2532-2535.

[8] B. Fatiha, B. Bouziane, and A. Baghdad, "MedIX: A named entity extraction tool from patient clinical reports." In Communications, Computing and Control Applications (CCCA), 2011 International Conference on, **(2011)**, pp. 1-6.

[9] H. Champion, N. Pizzi, and R. Krishnamoorthy, "Tactical Clinical Text Mining for Improved Patient Characterization." In Big Data (BigData Congress), 2014 IEEE International Congress on, **(2014)**, pp. 683-690.

[10] Y. Ling, X. Pan, G. Li, and X. Hu, "Clinical Documents Clustering Based on Medication/Symptom Names Using Multi-View Nonnegative Matrix Factorization", **(2015)**.

[11] Niazi, M. A. Khan, A. W. Muzaffar, M. Latif, and U. Qamar, "Signature automation of UMLS concepts: An un-supervised named entity recognition framework for classification of DNA and RNA in biological text." In Science and Information Conference (SAI), **(2015)**, pp. 727-733..

[12] Gu, Baohua, V. Dahl, and F. Popowich, "Recognizing biomedical named entities in the absence of human annotated corpora." In Natural Language Processing and Knowledge Engineering, 2007. NLP-KE 2007. International Conference on, **(2007)**, pp. 74-81.

[13] L. J. Gong, Y. Yuan, Y. B. Wei, and Xiao Sun, "A hybrid approach for biomedical entity name recognition." In Biomedical Engineering and Informatics, 2009. BMEI'09. 2nd International Conference on, **(2009)**, pp. 1-5.

[14] E. Apostolova, D. Channin, D. D. Fushman, J. Furst, S. Lytinen, and D. Raicu, "Automatic segmentation of clinical texts." In Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE, **(2009)**, pp. 5905-5908.

[15] X. Han and R. Ruonan, "The method of medical named entity recognition based on semantic model and improved SVM-KNN Algorithm." In Semantics Knowledge and Grid (SKG), 2011 Seventh International Conference on, , **(2011)**, pp. 21-27.

[16] L. Feng, X. Zhou, H. Qi, R. Zhang, Y. Wang, and B. Liu, "Development of large-scale TCM corpus using hybrid named entity recognition methods for clinical phenotype detection: An initial study." InComputational Intelligence in Big Data (CIBD), 2014 IEEE Symposium on, , **(2014)**, pp. 1-7.

[17] C. H. Wei, R. Leaman, and Z. Lu, "SimConcept: A Hybrid Approach for Simplifying Composite Named Entities in Biomedical Text", **(2015)**.

[18] A. Dehghan, J. Keane, and G. Nenadic, "Challenges in clinical named entity recognition for decision support." In Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, **(2013)**, pp. 947-951.

[19] T. Groza, A. Oellrich, and N. Collier, "Using silver and semi-gold standard corpora to compare open named entity recognisers." In Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on, , **(2013)**, pp. 481-485.

[20] M. Treins, O. Cure, and G. Salzano, “On the interest of using HL7 CDA release 2 for the exchange of annotated medical documents." InComputer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on, **(2006)**, pp. 524-532.

[21] E. W. Huang, T. L. Tseng, M. W. Chang, M. L. Pan, and D. M. Liou, “Generating standardized clinical documents for medical information exchanges." IT professional, vol. 12, no. 2, **(2010)**, pp. 26-32.

[22] S. DuVall, K. Boone, A. Gundlapalli, B. South, S. Shen, J. Nebeker, L. D'Avolio, and M. Samore, “Creating Reusable Annotated Corpora with the Clinical Document Architecture." In System Sciences (HICSS), 2011 44th Hawaii International Conference on, **(2011)**, pp. 1-10.

[23] C. H. Lin, W. S. Lai, L. H. Lee, H. M. Tsao, and D. M. Liou, “An entry generation pipeline for converting free-text medical document into Clinical Document Architecture document with entry-level." In Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on, **(2014)**, pp. 505-508.

Authors

Praveen Kumar, He received the B.E from the Department of Computer Engineering of University of Delhi, India in 2006. He is currently a Master course in Department of Computer Science of Lakehead University. His current research includes text analysis, NLP and DSL.

Sabah Mohammed, He received the M.S. degree in 1981 from Department of Computing of Glasgow University, UK and Ph. D. degree in 1986 from the Department of Computer Science of Brunel University, UK. Since 2001, he is working in the Department of Computer Engineering at Lakehead University as a Full Professor. Dr. Mohammed is also an adjunct Professor with the University of Western Ontario. His current research interests include Security of Health Data, Data Science, Cloud Computing, Social Networking and Enterprise Systems, Web-Based Systems, Mobile Computing and Big Data and Healthcare.

Arnold Kim, He received his MD in 1990 at the University of Western Ontario, and is an Assistant Professor with the Northern Ontario School of Medicine, Clinical Researcher at Thunder Bay Regional Health Sciences Center and Adjunct Professor with the Department of Computer Science of Lakehead University.

Jinan Fiaidhi, She is a full Professor and the Graduate Coordinator with the Department of Computer Science, Lakehead University, Ontario, Canada since late 2001. She is also an Adjunct Research Professor with the University of Western Ontario. She received her graduate degrees in Computer Science from Essex University (PgD 1983) and Brunel University (PhD, 1986). Dr. Fiaidhi research is focused on mobile and ubiquitous learning utilizing the emerging technologies (e.g. Cloud Computing, Calm Computing, Learning Analytics, Mobile Learning, Personal Learning Environment, Social Networking, Enterprise Mashups, Big Data and Semantic Web).