Identifying Medical Terms Related to Specific Diseases.pdf

Numerical Attribute Extraction.pdf

Identifying Medical Terms Related to Specific Diseases

Conference Paper · November 2015

DOI: 10.13140/RG.2.1.3659.3687

5 authors, including:

Sunil Mandhan

Hitachi, Ltd.

2 PUBLICATIONS **0** CITATIONS

SEE PROFILE

Raghavendra Ch

International Institute of Information Techno…

3 PUBLICATIONS **2** CITATIONS

SEE PROFILE

Kamalakar Karlapalem

International Institute of Information Techno…

186 PUBLICATIONS **2,448** CITATIONS

SEE PROFILE

Lini T Thomas

International Institute of Information Techno…

6 PUBLICATIONS **57** CITATIONS

SEE PROFILE

Identifying Medical Terms Related to Specific Diseases

Mihir Shekhar
IIIT Hyderabad, India
mihir.shekhar@research.iiit.ac.in

Veera Ragahvendra Chikka
IIIT Hyderabad, India
raghavendra.ch@research.iiit.ac.in

Lini Thomas
IIIT Hyderabad, India
lini.thomas@iiit.ac.in

Sunil Mandhan
Hitachi India Pvt. Ltd.,
Bangalore, India
sunilm@hitachi.co.in

Kamalakar Karlapalem[1]
IIT Gandhinagar, India
kkamal@iitgn.ac.in

***Abstract*—We present an automated disease term classification model using machine learning techniques that classifies a medical term to a specific disease class. We work on five particular diseases: Cancer, AIDS, Arthritis, Diabetes and heart related ailments. We identify medical terms specific to the five diseases including drugs, symptoms, abbreviations, disease names, tests, etc. We next classify the medical terms into specific disease classes. The results illustrate that our model for disease term classification finds all disease term classes with an average F-score of 0.966.**

***Keywords*—Medical Term Identification, Disease Term Classification, Machine Learning, Unified Medical Language System (UMLS)**

I. Introduction

Electronic Health Records or Discharge Summaries contain a lot of knowledge buried in the text. The increasing usage of electronic health records led to the need for effective natural language processing tools to extract the valuable information from these records. Information extraction from medical documents has caught much attention in the recent years. Electronic Health Record summarization driven by extracting symptoms, diagnoses, physical findings, test results, and therapeutic treatments has been done using information extraction methods [1]. The information extracted can be widely used in medical diagnosis improving the diagnostic speed, accuracy and as a support in training medical students. Automated systems have been developed to assist healthcare providers with quality assurance studies, or to support insurance processing. The availability of disease related profiles could be useful in dealing with diagnosis specific to the disease.

In this paper, we explore the automated acquisition of specific disease related knowledge buried in biomedical documents. Given a set of medical documents, we identify medical terms and associate them to the set of five specific diseases: Cancer, AIDS, Arthritis, Diabetes and heart related ailments. For example, the drug *Abacavir* is tagged with AIDS and *insulin* is tagged with Diabetes. Similarly the medical procedure *chemotherapy* is associated with Cancer, the test *blood glucose test* is associated with Diabetes, the abbreviation *CAD* is associated with heart related ailment, the symptom *myocardial ischemia* is associated with heart ailment, and *adenocarcinoma* is a type of Cancer.

Our system is more generic than previous works which finds association between specific medical entities like disease-drug or disease-symptoms using correlation based methods. Other forms of medical term association with disease terms like tests, procedure, treatment, synonyms, body location, abbreviations are also found by us. Correlation based methods suffer from its inability to do data prediction. The novelty of the paper lies in specific disease term prediction. The classifier predicts disease term with an average F-score of 0.82 (on overall diseases), for terms not present in the training data indicating good prediction ability of our method.

In order to identify terms related to each disease, our approach is to first locate all medical terms to include all disease related terms and then classify each medical term to a specific disease if associated. We next use a classifier to identify the disease tags of the terms which can be related to the specific five diseases. We employ four medical students to manually annotate the corpus of 136 discharge summaries of *ShARe/CLEF eHealth 2014* [2] and 1252 discharge summaries of *Informatics for integrating Biology and the Bedside 2009* (i2b2) [3]. Each medical term is either tagged as a generic term or is given a tag corresponding to one of the five diseases if strongly associated with it. We develop two separate modules, one for medical term identification and the other for disease term classification. The medical terms identified are fed to the second module wherein the terms that relate to the specific five diseases are identified. A brief statistics of the annotated corpus is given in Table I.

TABLE I. Database corpus Statistics

	CLEF	**i2b2**	**Combined**
Number Of Documents	136	1248	1384
Number Of Medical Terms	31701	273648	305349
Number Of Disease Terms	3164	42674	45838
AIDS	20	118	138
Cancer	173	967	1140
Diabetes	201	4542	4743
Heart	2711	36149	38860
Arthritis	59	898	957

[1]Work carried out at IIIT, Hyderabad

The contributions of this paper are

1) We create a medical term identifier that identifies the medical terms from the discharge summaries.
2) We create a disease term classifier that tags the medical terms strongly associated with particular disease
3) We run extensive experiments for finding the best set of features. We validate the efficiency of both models on various scenarios when tested individually and when functioning as one complete module.

In Section 2, we present the related work. In Section 3 we discuss our module for medical term identification. In Section 4 we present our work on disease term classification followed by results in Section 5. We conclude our work in Section 6.

II. Related Work

In clinical domain, Bio-Medical Named Term Recognition (BM-NER) plays an important role in getting a clear understanding of the large amount of clinical documents. Early works on biomedical literature was focused on rule based approaches. As the clinical information is growing at rapid pace, researchers have moved towards machine learning approaches like CRF [4] and SVM [5] for information extraction [6][7]. Different works of extracting information include identifying medications, tests, procedures [8], symptoms, protein names [9], enzyme Interactions and protein structures [10][11], etc, from clinical text. Other kinds of information extraction includes summarization of medical documents to a tabular format [12] by identifying the events, time and negativity. In [13] and [14] authors create disease-drug association rules from biomedical documents and patient record. [15] extracts the names of Genes and Gene products with a Hidden Markov Model. [14] extracts multiple rules for the diagnosis of coronary heart disease from a dataset which explicitly contains heart related data.

Two notable workshops which are engaged in organizing NLP research challenges in medical domain are Informatics for integrating Biology and the Bedside (i2b2) and Share/CLEF eHealth Evaluation Lab(CLEF). Major work on BM-NER started with the i2b2 challenges which targets information extraction from clinical text. The fourth i2b2 challenge 2010 focused on extraction and finding associations of medical terms related to disorders, tests, and treatments. The top performing systems in the i2b2 2010 challenge used statistically similar machine learning approaches with different features. The authors of [16] provide the state of art system of i2b2 2010 shared/task where they used semi-supervised approaches through clustering and is trained using CRFs with rich set of features of text, UMLS, cTAKES [17] and MedLEE NLP system [18] for extracting medical terms. [13] uses MedLEE to identify the medical term as disease or symptom and makes disease and symptom association pairs based on few statistical measurements. The other literature that relates though distantly to this problem is addressed in [16], [19] which categorizes medical terms to disease, treatment and tests. The work does not identify terms particular to a disease. [20] is one of the initial attempts to identify disease drug associations using correlation based methods to correlate a disease with a drug. The paper uses medLee to first find the list of all drugs and disease. Next, it uses correlation techniques given in [21], [22] to find

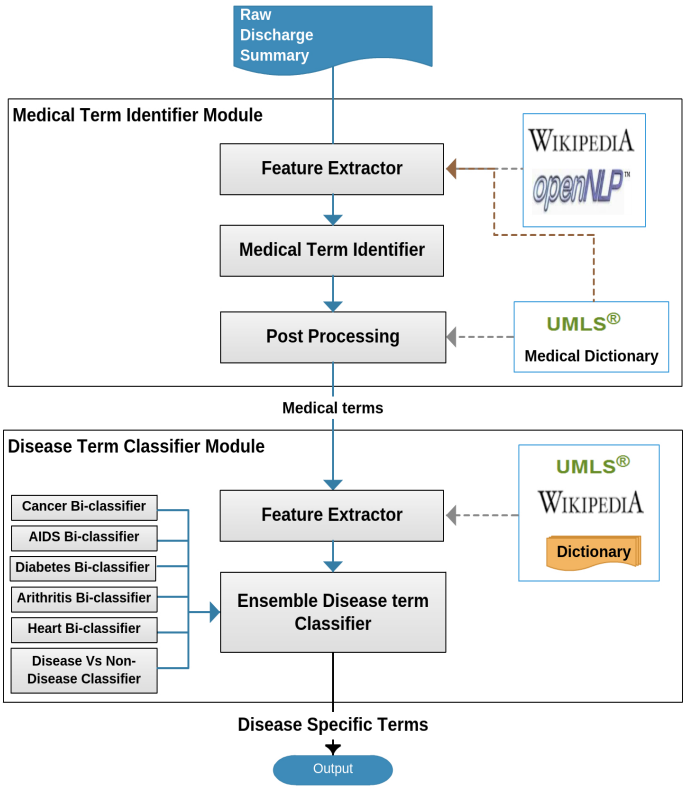

Fig. 1. Flowchart of Disease Specific Term Identification

correlations between disease names and drugs. In our work, we find associations of specific diseases to all generic forms associated with the disease including drugs, abbreviations, symptoms, synonyms, test, etc. Correlation based approaches would depend on the correlation to be present in the training data and hence is not suited for prediction.

The top five frequently seen diseases in our dataset have been selected for this work. It can be seen in Table I that the disease distribution is highly skewed with a very low representation for Arthritis and a high representation for heart ailments. We provide three experimental settings to test our results. One, the disease term classifier is tested independent of the medical term identifier model. This is done by feeding the gold standard medical terms to the disease term classifier as test input. Two, the test discharge summaries are first fed to the medical term identifier after which the medical terms identified are fed to the disease term classification. This experiment provides us with an insight to the pipeline setting shown in Figure 1. Third, we determine the prediction quality of our model by providing the disease term classifier with medical terms not present in the training corpus.

Figure 1 illustrates the sequence of steps in classifying disease related terms to specific disease classes given a discharge summary. It has two modules: (a) Medical term Identification and (b) Disease term Classification. Figure 1 briefly outlines the structure of both modules. In the next section we describe the medical term identification module, the features used and the results.

III. Medical Term Identification

Medical term identification is the essential first step in disease term classification. Since the disease terms that need to be given disease tags will be a subset of the medical terms found, it is essential that this module be developed with possibly high recall. Medical term identification has been attempted by different conferences on different datasets in the past decade. ShARe/CLEF eHealth 2013 focussed on identifying disease/disorder from the discharge summaries. The state-of-the-art system of this challenge [23] identifies disorder mention spans and their UMLS CUIDs with an F-score of 0.750. In addition to disease/disorder identification we also identify symptoms, tests, procedures, drugs, body locations and abbreviations. The medical term identifier attempts to find all medical terms that are associated with the five particular diseases. We shall refer to such a set of medical terms as disease related medical terms. Typical challenges of a medical term identifier is identifying abbreviations and noun phrases rarely encountered in the training set, misspelled terms, disjoint terms, out-of-vocabulary tokens, i.e., terms not found in the UMLS because of lack of coverage, etc [4]. We overcome these issues by using machine learning techniques like Conditional Random Fields (CRF) and Structured Support Vector Machines (SSVM) with rich set of features presented below.

A. Features Used

1) **Term feature :** Each space separated unigram (token) itself is used as a feature. The stemmed tokens are also used as a feature.
2) **POS tags :** Parts of speech of the word such as NN (noun), VB(Verb), PP(Preposition), etc. is used as a feature. Medical terms typically belong to a restricted set of POS tags which makes this feature useful. OpenNLP tagger was used for POS tagging [24].
3) **Chunk tags :** The parts of speech tags of a sentence are grouped into Chunk phrases. Similar to POS tag, each word has a chunk tag. Few chunk tags include B-NP (Beginning of Noun Phrase), I-NP (Intermediate term of Noun Phrase), B-VP (Beginning of Verb Phrase), etc. The chunk tags play an important role in the identification of boundary of the medical term.
4) **Prefix and Suffix :** Many medical terms share common prefixes and suffixes. Few examples of prefix are cardio-, dia-, exo-, gluco-, glyc-, etc, and suffix are -sis, -logy, -fusion, -globin, -ium, -ist, etc which are strongly associated with the medical terms present in discharge summary.
5) **UMLS Feature :** Unified Medical Language System (UMLS) contains the medical vocabularies gathered from different sources, making it one of the largest knowledge bases of medical terminology. This boolean feature checks the existence of the token in UMLS.
6) **UMLS Semantic type :** In UMLS all the terms are categorised into various semantic types[1] for better understanding of the term. Certain medical terms can not correspond to any specific disease. For example *mg, discharge summary, bed, operation theatre, hospital,* etc are not specific to any disease. Thus, a subset of the semantic types that represent disease specific medical terms are used [25].
7) **Orthographic feature :** Orthographic feature refers to the detailing around the structure of the word. The Orthographic feature contains details regarding the word structure which include information regarding the word starting with an upper case letter or a lower case letter or whether the word is completely in upper case or lower case. Further information about the presence of hyphen, underscore, etc in the word are also captured.
8) **Stopword :** This is a binary feature used to check whether the given token is a stopword or not.
9) **Wikipedia feature :** Wikipedia [26] is used to get more information about the word. We index wiki page title and description using lucene. We use the wiki page title and the wiki categories obtained as search result after querying the term, as a feature.
10) **Medical dictionary :** A medical dictionary is created using training dataset comprising of all the medical terms annotated in the data. This binary feature is used to check whether the word occurs in the dictionary or not.
11) **Section Header :** A discharge summary is often divided into different sections. Few section headers include: History of Present Illness, Physical Examination, Past Medical History, Procedure, Discharge Medications, etc. Most of the medical terms would frequently fall under specific sections. Hence, section header is used as a feature in identifying medical term.

TABLE II. Comparison of CRF and SSVM Model for Medical Term Extraction

		Medical Terms		Disease Terms	
		CRF	SSVM	CRF	SSVM
Strict	Precision	0.868	0.806	0.143	0.132
	Recall	0.814	0.761	0.892	0.835
	Fscore	0.840	0.783	0.246	0.229
Relaxed	Precision	0.954	0.934	0.1525	0.151
	Recall	0.894	0.882	0.951	0.951
	Fscore	0.923	0.908	0.262	0.261

B. Approach

We use both Conditional Random Fields (CRF) [27] and Structured Support Vector Machines (SSVM) [28] over the features listed above to build our machine learning tool. The discharge summaries in CLEF and i2b2 were fed together for training and testing. Various experiments conducted showed that the CRF based model gave better results as compared to SSVM. Results pertaining to a strict evaluation and a relaxed evaluation have been presented for the CRF and SSVM model in Table II. In the results pertaining to " Strict Evaluation", the medical term identified is identical to the reference gold standard term. In "relaxed evaluation", the medical term overlaps with the reference gold standard term. Consider the sentence *"The patient had a history of insulin dependent diabetes mellitus "*. Let "insulin dependent diabetes mellitus" be tagged as the medical term in gold standard and let "diabetes mellitus" be the medical term identified by our model. Relaxed evaluation accepts the identified term

[1] http://metamap.nlm.nih.gov/Docs/SemGroups_2013.txt

TABLE III. MEDICAL TERM EXTRACTION RESULTS ON IDENTIFYING MEDICAL TERMS BEFORE AND AFTER POST-PROCESSING

		Before Post Processing			After Post Processing		
		CLEF	i2b2	Combined	CLEF	i2b2	Combined
Strict	Precision	0.829	0.873	0.868	0.769	0.786	0.784
	Recall	0.751	0.821	0.814	0.782	0.856	0.849
	F-score	0.788	0.846	0.840	0.776	0.819	0.815
Relaxed	Precision	0.959	0.954	0.954	0.892	0.859	0.862
	Recall	0.869	0.897	0.894	0.908	0.936	0.933
	F-score	0.912	0.925	0.923	0.900	0.896	0.896

TABLE IV. MEDICAL TERM EXTRACTION RESULTS ON IDENTIFYING DISEASE RELATED MEDICAL TERMS BEFORE AND AFTER POST-PROCESSING

		Before Post Processing			After Post Processing		
		CLEF	i2b2	Combined	CLEF	i2b2	Combined
Strict	Recall	0.844	0.896	0.892	0.866	0.926	0.922
Relaxed	Recall	0.952	0.951	0.951	0.983	0.987	0.986

as the medical term as it overlaps with the gold standard medical term, whereas strict evaluation will not accept the identification as correct as it is not an exact match.

We perform a four fold cross validation to obtain the results presented. It can be seen from Table II that the CRF model outperforms the SSVM model in medical term identification and medical terms that belong to one of the diseases as well. Hence, the CRF model has been used in medical term identification. Since the disease term classifier would depend on the medical term identifier module, we make further efforts to improve the recall of the CRF model using post processing steps. It can be noted that while the recall of medical terms and disease terms in Table II is reasonable, the precision of the disease terms is very low. The medical term classifier was built attempting to maximise the recall of medical terms and disease terms by the classifier.

When a discharge summary is fed to the medical term identifier module, the text first undergoes a feature extraction phase using Wikipedia, UMLS, NLP methods and medical dictionary as seen in the Figure 1. The CRF based medical term identifier then identifies the medical terms. Discharge summary also undergoes a post processing phase described below, before being fed to the disease term classifier module.

The post processing step is done in order to increase the recall of the medical terms pertaining to some disease. We use a rule-based approach, to capture the medical terms that have not been identified by our model and to extend the boundary of partially recognized medical terms. The following three techniques are used to increase the recall of the disease related medical terms.

1) Medical dictionary (which was created from training dataset) is used to tag the medical terms in the test data if not identified by the machine learning model.
2) UMLS dictionary is used to identify the medical terms which were not identified by the machine learning model. The medical terms identified are removed from the discharge summary. The n-grams (4-grams and smaller n-grams) extracted from the remaining text are fed to UMLS. The longest n-gram present in UMLS is tagged as a medical term in the test data. For example, let us consider the sentence, "This patient is a 72-year-old woman with a significant ischemic heart disease ". Out of the medical terms "heart disease " and "ischemic heart disease" which are related to Heart, we extract the longest n-gram "ischemic heart disease" as the medical term.
3) The test data contains medical terms with multiple occurrences. It was seen that while certain occurrences of the medical terms was identified by our model, other occurrences remained unidentified. Hence, for each medical term found by our model, all occurrences of the medical term in the test data are also tagged as a medical term. In cases of conflict, the largest matching term was used.

Table III gives details of the results of medical term identification using the CRF model before and after post processing when run on the two datasets. The relaxed and strict evaluation results of both datasets is presented individually and together. It can be seen that the post processing step has increased the performance of medical term identifier module. The precision of the system shows a slight decrease due to repetitive tagging of false positives in this approach. It will be shown in the following section that the disease term classifier is robust with a strong set of features that gives good performance in spite of the decreased precision. The medical term identifier module attempts to maximize the recall of the disease related medical terms. Table IV shows the recall of the disease related medical terms extracted before and after post-processing. It can be clearly seen that the recall of the disease related medical terms has increased which will subsequently improve the performance of the disease classifier module.

IV. DISEASE TERM CLASSIFICATION

In this section we discuss the model for disease term classification, its challenges, features used, experimentation done, followed by results in the next section. The disease term classification model results depend on the recall of the disease related medical terms extracted by the medical term identifier. We show results both when the gold standard medical terms are fed as test data to the disease term classifier and when the medical term extracted by the model in Section 3 is fed as test data. It can be seen that the results improve significantly

when gold standard medical terms are used in the test data. This implies if the module in Section III is replaced by a model of higher accuracy then the results for disease term classification will automatically increase.

TABLE V. MULTICLASS DISEASE TERM CLASSIFIER

	Cancer	AIDS	Diabetes	Heart	Arthritis
Recall	0.9020	0.3514	0.9511	0.9320	0.9404
Precision	0.9368	1.0	0.9951	0.9914	1.0
F-score	0.9191	0.52	0.9726	0.9608	0.9693

We use support vector machines to build our model. Support Vector Machines are widely used in natural language processing tasks because of its ability to handle extremely large feature set efficiently in spite of feature sparsity which is common in NLP tasks. The manually annotated corpus was divided into partitions of 70% and 30%. The first partition is used for training and the latter is used for testing. We first trained and tested a multi-class SVM whose results are presented in Table V using LIBSVM [29] and the RBF [30] kernel was used with all the parameters set to default. The feature space of the RBF kernel extends to infinite dimensions. Thus, the RBF kernel can project a feature to infinite dimensions and hence it can generate a decision boundary between two classes for extremely high dimensions efficiently. Class size imbalance poses a challenge in building the multiclass classification. The distribution of the disease tags with respect to the different diseases is highly skewed as can be seen in Table I. For example Heart related terms in i2b2 are 36,149 while the number of Cancer terms are 967. Also, the ratio of the number of disease terms to non-disease medical terms is 1:10. Poor representation of the terms of diseases like Arthritis and AIDS resulted in low recall when a multiclass SVM classifier was used as in Table V. The features used are same as listed in IV-A below.

In order to address the issues that arise from class size imbalance among the disease classes, we train a bi-class SVM classifier with a RBF kernel for each disease where the classifier classifies a medical term to the specific disease if associated with it. Cost factor C in libsvm is adjusted to give higher weightage to rare classes in order to achieve a balance. Different cost factors have thus been used for each of the five bi-classifiers. The Figure 1 shows the details of the disease term classifier module. An ensemble approach is then used to build the complete disease term classifier. The disease term classifier is trained over the five bi-class classifiers and further results of these bi-classifiers are merged along with the results of disease vs non-disease classifier using the Naive-Bayes classifier. On comparing the results of the multi-class classifier given in Table V and the bi-class classifier given in Table VII, it can be seen that the bi-class classifier performs better. The medical terms identified by the medical term identifier module are fed to the disease term classifier module which first extracts the features of the medical terms using Wikipedia, UMLS and disease specific dictionaries as shown in Figure 1. The ensembled disease term classifier uses the features extracted and classifies the medical terms into disease specific classes or the generic class.

The following subsection presents the experimentation with feature selection, the effect of various features selected, the challenges and importance of the feature set finally used.

A. Features Description

We experimented with following features for building the disease classification model.

1) **Term Feature:** Each medical term obtained from the training data is tokenized on whitespace character and each token was stemmed. An n-gram medical term is represented by the corresponding unigrams. This feature is used solely for the purpose of medical term representation.
2) **UMLS Feature:** The UMLS page text corresponding to each medical term was tokenized on white-space characters after stopword removal. Each tokenized term was stemmed to create the vocabulary of the UMLS feature. This feature was included with the intuition that a UMLS page corresponding to a term related to a particular disease will contain other relevant terms particular to the disease which will further help identify the disease.
3) **Proximity Feature:** The unigram appearing before and after the medical term is used as this feature. In case of ambiguous medical terms, the unigrams before and after the medical term could suggest information regarding the association of the medical term to a particular disease. On experimentation it was concluded that the proximity feature is not useful since the ambiguous medical terms were identified by the other features. Also, the quality of results was seen to fall on including the proximity feature.
4) **Dictionary Feature(Dict):** A bag of words containing root words and some common drugs forms this feature. If a unigram of medical term contains or is contained in an entry of the dictionary feature, then, the medical term is assigned to the corresponding disease tag in this feature. Hence, the n-gram of a given medical term might be related to multiple entries of the dictionary.
5) **Semantic Type Feature:** The feature represents the UMLS semantic types of the medical terms.
6) **Wikipedia features(W):** A wiki page corresponding to a medical term would contain several other medical terms closely associated with it. Hence, finding pages relevant to the medical term in Wikipedia was found to help associate the term to the respective disease it relates to. A Lucene index was created on the Wikipedia dump [26]. The content and the categories of the top ten pages are used to build the Wiki feature set.

The term features and the semantic features in combination form the baseline features represented by **B** in the Table VII.

B. Feature Pruning and Selection

The term feature and semantic type feature together as mentioned in Section IV-A are used as baseline features. We conducted extensive experimentation to identify best subset of features which was further used to generate the final biclassifier models. Table VI presents the intermediary results at each level of feature pruning. As can be observed from

TABLE VI. SIZE OF FEATURE SET BEFORE AND AFTER PRUNING

	Original	Null Removed	Pruned using Info Gain	Percentage Reduction
Baseline	8718(8585+133)	5877(5744+133)	5102(5001+101)	58.5
UMLS	8939	7611	5001	55
Wikipedia	1181568(137593+1043975)	74159(40156+34003)	23002(20001+3001)	1.94
Proximity	30992	5396	4001	12.909
Dictionary	200	–	—	—

TABLE VII. INFLUENCE OF THE FEATURE SET

	B	B+ UMLS	B+W	B+W+ UMLS	B+W+ UMLS+Dict	B+W+UMLS+ Dict+Proximity
	Recall (Precision)	Recall (Precision)	Recall (Precision)	Recall (Precision)	Recall (Precision)	Recall (Precision)
Cancer	0.8581(0.9270)	0.9020(0.9852)	0.9087(0.9711)	0.8986(0.9708)	**0.9088(0.9746)**	0.9088(0.9746)
AIDS	0.5675(0.875)	0.7297(0.9)	0.8649(1.0)	0.8649(1.0)	**0.8649(1.0)**	0.8649(1.0)
Diabetes	0.9434(0.9967)	0.9589(0.9976)	0.9535(0.9935)	0.9558(0.9952)	**0.9891(0.9938)**	0.9891(0.9930)
Heart	0.9173(0.9902)	0.9344(0.9947)	0.9528(0.9905)	0.9566(0.9904)	**0.9666(0.9908)**	0.9674(0.9906)
Arthritis	0.9508(0.9855)	0.9368(0.9889)	0.9439(0.9890)	0.9439(0.9853)	**0.9474(0.9854)**	0.9509(0.9855)
DiseaseVsNonDisease	0.6569(0.9989)	0.7566(0.9937)	0.8982(0.9886)	0.9315(0.9902)	**0.9315(0.9904))**	0.9297(0.9902)

Table VI, original feature size was very large for each of the features mentioned in Section IV-A and thus feature pruning was an essential step. We first removed all the features with null values. Further pruning was done using information gain criteria. Information gain [31] is used in decision trees to define a preferred sequence of attributes to choose the parent node. This technique has been found to be very effective in feature selection in textual data [31]. The Table VI details feature size before and after pruning. The size of pruned feature was also fixed using cut-off threshold on information gain.

Feature set selection is done by experimenting with the set of features by adding the pruned feature set one at a time and comparing the efficiency of the model. The gold test data was used for evaluating various models. For a better understanding of the effect of features on accuracy, we also create a new classifier which classifies disease terms from the non-disease terms. Thus it gives an idea of how features help in classifying the disease and non-disease terms. This classifier is also used in building the ensembled classifier which will be explained later. While choosing the feature set, we take into consideration the effect on accuracy of this classifier as well as other five biclasssfier corresponding to diseases Cancer, AIDS, Diabetes, Arthritis and heart related ailments. Table VII presents the recall and precision obtained on test data as the feature set is varied to include features one by one.

A steady increase in recall and precision can be seen for almost all the classifiers as features set is incremented to become Baseline + UMLS and Baseline + Wiki. Further when we use Baseline + Wiki+ UMLS features the precision and recall of cancer model marginally decreases. But we can see there is overall increment in recall and precision for the other disease models. The best accuracy is given by Baseline + Wiki + UMLS + Dict. On further adding the proximity feature, a detrimental or null effect on recall and precision of the different diseases is observed. Moreover there is a sharp decline in accuracy numbers of disease vs non disease classifier indicating these features are actually overall reducing the efficiency of the model to classify disease related medical terms (corresponding to specific diseases) vs non-disease term. Hence Baseline + Wiki + UMLS + Dict feature set are used to build the final model. In this text if not otherwise stated, this feature set has been used for all the experiments conducted. This experiment also demonstrates the efficiency of the model independent of the medical term classifier where it is assumed that medical term classifier would find all medical terms correctly.

V. RESULTS

We present experimental results on the two datasets CLEF 2014 and i2b2 2009. The clef and i2b2 datasets have been divided into two parts, 70% for training while the remaining 30% is used as test data. We use recall, precision and F-score to present the efficiency of the model.
The recall, precision and F-score of each bi-class classifier trained by us for each disease can be calculated using the following formulas where A represents the label of a specific disease class:

Recall for label A = $\frac{TP_A}{TP_A+FN_A}$

Precision for label $A = \frac{TP_A}{TP_A+FP_A}$

F-score for label $A = \frac{2\times Recall\times Precision}{Recall+Precision}$

where TP_A refers to True Positives of label A, FN_A refers to false negatives of label A, FP_A refers to False Positives of label A. Also $(TP_A + FN_A)$ is equal to total number of gold terms with label A and $(TP_A + FP_A)$ is equal to total number of terms predicted with label A
It is evident that the accuracy of the disease term classification model is dependent on the results of the medical term identification module. We further experiment with three scenarios.

1) The experiment evaluates the efficiency of the disease term classifier. We provide as input to the disease term classifier, all the medical terms in the gold test data. The classified disease terms are compared with the original disease tags in the gold test data. Thus in this scenario the disease tag classifier is independent of the medical classifier model. The

results can be seen in Table VII (presented in bold) which corresponds to feature set B+ UMLS + Wiki + Dict. It was seen that an average F-score of 0.9663 was achieved over the five diseases.

2) Given discharge summaries as input, we first run our medical term identification module over the discharge summaries. The results of the medical term identifier are then fed to the disease classifier. The results so obtained are compared with gold test data. Section V-A describes in detail this experiment where the medical term identifier and disease term classifier are pipelined.
3) This experiment evaluates the prediction quality of our disease term classifier. We feed the disease term classifier with medical terms not present in the training set. Subsection V-B presents the details and results of the experiment.

A. Pipelined experiment with medical term identifier and disease term classifier

The results presented in Table VII are independent of the performance of the medical term identifier discussed in Section III. The medical term identifier developed by us gives a recall of 0.92. This limits the performance of our disease term identifier as not all disease related medical terms are found. In this experiment, whose result is presented in Table VIII, we give the performance results of the disease term classifier when fed with the results of a medical term identifier. The abbreviations used in the table imply the below:

1) **P, R, F:** Overall precision, recall and F-score of the model as output by model 2 with respect to the gold.
2) **G:** Number of disease tags of the particular disease in gold.
3) **M1:** Number of disease tags of the particular disease present among the medical terms found by the medical term classifier.
4) **M2:** Number of disease tags of the disease correctly identified by the disease term classifier, which are the true positives TP.
5) **FT:** Number of medical terms marked as disease tag of the particular disease (FP+TP)

TABLE VIII. DISEASE TERM CLASSIFIER ON THE RESULTS OF MEDICAL TERM IDENTIFIER

	Cancer	**AIDS**	**Diabetes**	**Heart**	**Arthritis**
Recall	0.8571	0.65	0.8130	0.7757	0.8136
Precision	0.82076	0.65	0.8220	0.71683	0.6486
F-score	0.838	0.65	0.8175	0.7451	0.7218
G	203	19	1192	9976	235
M1	182	17	999	8231	199
M2	174	13	970	7739	192
FT	212	19	1179	10795	295

In Table VIII the rows marked as "M1" and "M2" show the particular disease terms that were made available to the disease term classifier and the disease tags correctly assigned. AIDS has the lowest F-score. This can be attributed to the fact that the number of the terms of AIDS is very low. Heart though maximally represented has a F-score lower than that of Diabetes. It was seen that Diabetes contains a limited number of unique terms associated with it most of which was the term 'insulin' or variations of the term 'insulin'. Heart related terms however have a huge variation where the number of unique heart terms is 1307 while the number of unique Diabetes terms is 212.

B. Test Data with terms not present in Training Data

The training data contains several medical and disease terms that overlap with the terms in the test data. However, the prediction capability of the model can be best tested if fed with terms not present in the training data. We form test data by removing all medical terms that are present in the training data. Out of 75355 medical terms present in the test data, it was seen that 44834 terms were not present in the training data. These 44834 terms were fed as input test data to the disease term classifier. Table IX presents the results on the experiment. AIDS shows a low F-score values, since the representation of AIDS is very low in the training data with only 138 AIDS related terms present among the 45,838 disease terms. Further, 11 out of 18 AIDS related terms present in the test data are not predicted in this experiment resulting in the low recall. It was found that the 11 terms not predicted were repetitions of the only two drugs nevirapine (count : 3) and zidovudine (count : 8). The experiment shows good F-score values for all the other four diseases which suggests a good classifier capacity of the classifier developed.

TABLE IX. TEST DATA WITH TERMS IN TRAINING DATA REMOVED

	Cancer	**AIDS**	**Diabetes**	**Heart**	**Arthritis**
Recall	0.9127	0.38888	0.8102	0.8077	0.9467
Precision	1.0	1.0	1.0	1.0	0.7158
F-score	0.9543	0.56	0.8951	0.8936	0.8152

VI. CONCLUSION

In this work, we identify terms related to specific five diseases being considered. Two models are created for the purpose where the first classifier attempts to identify all medical terms, thus finding all the disease related medical terms which will be classified later to particular diseases. The next classifier classifies the medical terms into the five specific diseases if associated, namely AIDS, Cancer, Diabetes, Heart related terms and Arthritis. From experiments it was found that the CRF model for medical term identification and the SVM biclass classifier model for disease term classification are best suited. Various experimental results have been presented to demonstrate the efficiency and utility of the system developed. The paper provides a generic method that can be adopted for finding disease tags specific to any set of diseases.

REFERENCES

[1] M. S. Simpson and D. Demner-Fushman, "Biomedical text mining: a survey of recent progress," in *Mining text data*. Springer, 2012, pp. 465–517.

[2] CLEF2014, "Conference and labs of the evaluation forum 2014." [Online]. Available: http://clef2014.clef-initiative.eu/

[3] i2b22009, "Informatics for integrating biology and the bedside 2009." [Online]. Available: https://www.i2b2.org/NLP/Medication/

[4] A. Bodnari, L. Deléger, T. Lavergne, A. Névéol, and P. Zweigenbaum, "A supervised named-entity extraction system for medical text," in *Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013.*, 2013. [Online]. Available: http://ceur-ws.org/Vol-1179/CLEF2013wn-CLEFeHealth-BodnariEt2013.pdf

[5] M. A. Andrade and A. Valencia, "Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families," *Bioinformatics*, vol. 14, no. 7, pp. 600–607, 1998. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/14.7.600

[6] N. Johri, Y. Niwa, and V. R. Chikka, "Optimizing apache ctakes for disease/disorder template filling: Team hitachi in the share/clef 2014 ehealth evaluation lab." in *CLEF (Working Notes)*, 2014, pp. 111–123.

[7] V. R. Chikka, N. Mariyasagayam, Y. Niwa, and K. Karlapalem, "Information extraction from clinical documents: Towards disease/disorder template filling," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, ser. Lecture Notes in Computer Science, J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. J. F. Jones, E. SanJuan, L. Cappellato, and N. Ferro, Eds., vol. 9283. Springer, 2015, pp. 389–401. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-24027-5_41

[8] S. Doan and H. Xu, "Recognizing medication related entities in hospital discharge summaries using support vector machine," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, ser. COLING '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 259–266. [Online]. Available: http://dl.acm.org/citation.cfm?id=1944566.1944596

[9] S. Dingare, M. Nissim, J. Finkel, C. Manning, and C. Grover, "A system for identifying named entities in biomedical text: how results from two evaluations reflect on both the system and the evaluations: Conference Papers," *Comp. Funct. Genomics*, vol. 6, no. 1-2, pp. 77–85, 2005. [Online]. Available: http://dx.doi.org/10.1002/cfg.v6:1

[10] K. Humphreys, G. Demetriou, and R. Gaizauskas, "Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures," 2000.

[11] R. Gaizauskas, G. Demetriou, and K. Humphreys, "Term recognition and classification in biological science journal articles," in *In Proc. of the Computional Terminology for Medical and Biological Applications Workshop of the 2 nd International Conference on NLP*, 2000, pp. 37–44.

[12] E. Aramaki, Y. Miura, M. Tonoike, T. Ohkuma, H. Mashuichi, and K. Ohe, "Text2table: Medical text summarization system based on named entity recognition and modality identification."

[13] X. Wang, A. E. Chused, N. Elhadad, C. Friedman, and M. Markatou, "Automated knowledge acquisition from clinical narrative reports," in *AMIA 2008, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 8-12, 2008*. AMIA, 2008. [Online]. Available: http://knowledge.amia.org/amia-55142-a2008a-1.625176/t-001-1.626020/f-001-1.626021/a-162-1.626076/a-163-1.626073

[14] A. Hara, T. Ichimura, and K. Yoshida, "Discovering multiple diagnostic rules from coronary heart disease database using automatically defined groups." *J. Intelligent Manufacturing*, vol. 16, no. 6, pp. 645–661, 2005.

[15] N. Collier, C. Nobata, and J.-i. Tsujii, "Extracting the names of genes and gene products with a hidden markov model," in *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, ser. COLING '00. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 201–207. [Online]. Available: http://dx.doi.org/10.3115/990820.990850

[16] B. de Bruijn, C. Cherry, S. Kiritchenko, J. D. Martin, and X. Zhu, "Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010," *JAMIA*, vol. 18, no. 5, pp. 557–562, 2011. [Online]. Available: http://dx.doi.org/10.1136/amiajnl-2011-000150

[17] 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," *Journal of the American Medical Informatics Association*, vol. 17, no. 5, pp. 507–513, 2010.

[18] C. Friedman, "Medlee-a medical language extraction and encoding system," *Columbia University, and Queens College of CUNY*, 1995.

[19] R. G. Parth Pathak and G. Joshi, "Crf-based clinical named entity recognition using clinical nlp features."

[20] E. S. Chen, G. Hripcsak, H. Xu, M. Markatou, and C. Friedman, "Automated acquisition of disease–drug knowledge from biomedical and clinical documents: an initial study," *Journal of the American Medical Informatics Association*, vol. 15, no. 1, pp. 87–98, 2008.

[21] H. Cao, M. Markatou, G. B. Melton, M. F. Chiang, and G. Hripcsak, "Mining a clinical data warehouse to discover disease-finding associations using co-occurrence statistics," in *AMIA Annual Symposium Proceedings*, vol. 2005. American Medical Informatics Association, 2005, p. 106.

[22] H. Cao, G. Hripcsak, and M. Markatou, "A statistical methodology for analyzing co-occurrence data from a large sample," *Journal of biomedical informatics*, vol. 40, no. 3, pp. 343–352, 2007.

[23] B. Tang, Y. Wu, M. Jiang, J. C. Denny, and H. Xu, "Recognizing and encoding discorder concepts in clinical text using machine learning and vector space model," in *Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013.*, 2013. [Online]. Available: http://ceur-ws.org/Vol-1179/CLEF2013wn-CLEFeHealth-TangEt2013.pdf

[24] J. Baldridge, "The opennlp project," *URL: http://opennlp. apache. org/index. html,(accessed 2 February 2012)*, 2005.

[25] "Gold umls semantic types." [Online]. Available: http://cde.iiit.ac.in/goldUMLSSemanticTypes.txt

[26] Wikipedia, "Wikipedia dump 10 2014." [Online]. Available: http://dumps.wikimedia.org/enwiki/20141008/

[27] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[28] T. Joachims, "Sequence tagging with structural support vector machines."

[29] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.

[30] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[31] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *ICML*, vol. 97, 1997, pp. 412–420.

Numerical Atrribute Extraction from Clinical Texts

RESEARCH · NOVEMBER 2015

DOI: 10.13140/RG.2.1.4763.3365

1 AUTHOR:

Sunil Mandhan

Hitachi, Ltd.

2 PUBLICATIONS **0** CITATIONS

Extraction and Association of Numerical Attributes and Values from Electronic Health Records

[1]Sarath P R, [1]Sunil Mandhan, and [2]Yoshiki Niwa

[1]Research and Development Centre, Hitachi India Pvt. Ltd., Bangalore, India
{sarath,sunilm}@hitachi.co.in
[2]Hitachi Ltd., Central Research Laboratory, Japan
yoshiki.niwa.tx@hitachi.com

Abstract. This paper describes about information extraction system, which is an extension of the system developed by team Hitachi for "Disease/Disorder Template filling" task organized by ShARe/CLEF eHealth Evolution Lab 2014. In this extension module we focus on extraction of numerical attributes and values from discharge summary records and associating correct relation between attributes and values. We solve the problem in two steps. First step is extraction of numerical attributes and values, which is developed as a Named Entity Recognition (NER) model using Stanford NLP libraries. Second step is correctly associating the attributes to values, which is developed as a relation extraction module in Apache cTAKES framework. We integrated Stanford NER model as cTAKES pipeline component and used in relation extraction module. Conditional Random Field (CRF) algorithm is used for NER and Support Vector Machines (SVM) for relation extraction. For attribute value relation extraction, we observe 95 % accuracy using NER alone and combined accuracy of 87 % with NER and SVM.

Keywords: NLP, NER, relation extraction, information extraction, crf, svm

1 Introduction

Healthcare providers are increasingly adopting Electronic Health Record (EHR) systems to improve the quality of care. Nowadays EHR data and systems are accessible to patients (patient portals) and non-expert clinical professionals. Clinical information inside EHR systems are various and mainly are in the form of unstructured text (e.g. Discharge Summary). It is difficult for non-expert end users to interpret the documents which contain many medical abbreviations and jargons [8]. Extracting frequently required information from unstructured clinical text and representing in a structured manner will give quick and timely access to patient's health related data to end users. For example information about patient's body vital signs, blood components, drugs etc. are used in day to day operations to understand the progress and treatment of patient. Most of such information is in numerical form. In this paper we describe about the experiment, developed system and results of numerical attributes and related values extraction from discharge summary records.

1.1 Problem Description

Attributes are originated from physical examinations and medical tests required for disease diagnosis as well as treatment procedures. For example, blood pressure and heart rate are the common and important numerical measurements required for diagnosis of almost all the diseases. Table 1 shows some important numerical attributes which are prominent in clinical diagnosis and found in discharge summaries [8].

Table 1. Types and examples of numerical attributes

Class	Examples of attributes
Vital Signs	Blood pressure, temperature, pulse, heart rate, respiratory rate, oxygen saturation
Blood Components	WBC, RBC, hematocrit, platelets count, glucose, urea, nitrogen, sodium, potassium, anion gap
Drug Attributes	Dosage, quantity, frequency, periodic interval

Example Sentence. "Her vital signs the following day, she had heart rate of 66, blood pressure 120/63, respiratory rate 14, 100% on 5 liters nasal cannula O2 saturation."

Manual Annotation. "Her vital signs the following day, she had <attribute-1>heart rate<attribute-1> of <value-1>66<value-1>, <attribute-2>blood pressure</attribute-2> <value-2>120/63</value-2>, <attribute-3>respiratory rate</attribute-3> <value-3>14</value-3>, <value-4>100%</value-4> on 5 liters nasal cannula <attribute-4>O2 saturation</attribute-4>."

2 Related Work

There are many references available for extracting different types of information from clinical documents. Identifying medications, tests, procedures [2], symptoms, protein names [3], enzyme interactions and protein structures [4] are some examples.
Other kinds of information extraction include summarization of medical documents to a tabular format by identifying the events, time and negativity [9].

I2B2 [5] and CLEF [6] are the notable workshops which are engaged in organizing NLP research challenges in medical domain. The CLEF paper [1] has used SVM for relation extraction for detecting relationship between disease and different properties of disease like body location, severity, etc. SVM was used in case where more than one body locations were present to establish the relationship which body location is associated to the disease.

To the best of our knowledge, we could not find any related work where numerical attributes and values extraction from clinical documents is designed and evaluated as a relation extraction problem using a combination of supervised machine learning techniques. In this paper, we attempt to solve this problem and explain the different algorithms with input features and relevant data pre-processing.

3 System Architecture

We solve the problem in two steps. First step is extraction of numerical attributes and values, which is developed as a Named Entity Recognition model using Stanford NLP libraries. Second step is correctly associating those attributes to values, which is developed as a relation extraction module in Apache cTAKES framework. We integrated Stanford NER model as a cTAKES pipeline component and used in relation extractor. Figure 1 describes the architecture of the developed system. Section 3.1 describes about algorithm and approaches in attributes and values extraction. Section 3.2 describes about relation extraction module.

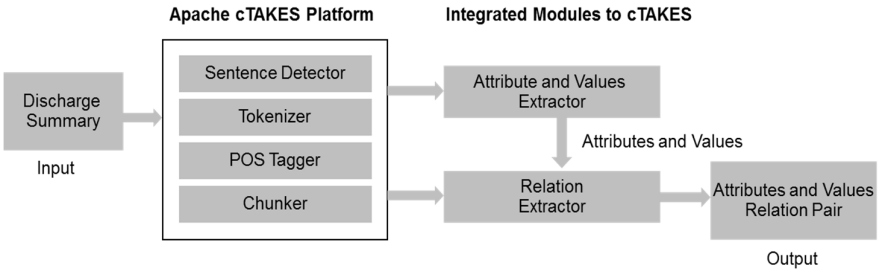

Fig. 1. System Architecture

3.1 Attributes and Values Extraction

We trained a model for Attribute and Value extraction from discharge summaries using Stanford NER library. Stanford NER provides a general implementation of arbitrary order linear chain Conditional Random Field (CRF) sequence models [10]. It can be trained for any task.

3.1.1 Tokenization: We observed the default Stanford Tokenizer (Penn Treebank Tokenizer) splits certain attribute words in discharge summary in an undesired manner as explained with an example below.

Word	Required Tokenization	Output of Stanford Tokenization
WBC-12.8*#	WBC, - , 12.08, *, #	WBC-12, .8, *, #

To avoid this issue we applied a regular expression based preprocessing before input data for Tokenization. We replaced the hyphens in certain types of attributes with white space as it is harmless to do so. Following regular expression is used for preprocessing.

Regular expression: ([a-z A-Z]|O3|O2|B12)(-)([0-9])

After preprocessing, we get the following tokenization

Word	Required Tokenization	Output of Stanford Tokenization
WBC 12.8*#	WBC, 12.08, *, #	WBC,12.8, *, #

Modification to the regular expression or new implementation of tokenizer is applicable as and when more issues are detected in tokenization.

3.1.2 Model Training: From the tokens and our manually annotated discharge summaries, we programmatically prepared training data for Stanford CRF Classifier. We used the following features from Stanford NER Feature Factory [11] for training the CRF model.

1. Word
2. Position (word index in sentence)
3. Word shape feature
4. Ngrams from word
5. Disjunctions of words

3.2 Relation Extraction

SVM algorithm is used for establishing the relationship between attribute and value. SVM is a distance based method and has proved to be effective for relation extraction [1]. The basic idea of using SVM on relationships is to map a relation into a feature space and find the maximum margin hyper plane to separate two classes (related and not related).

3.2.1 Model Training: We trained SVM model for relation association using manually annotated discharge summaries. Following features are used.

1. Part of speech
2. Punctuation
3. Phrase chunking (Noun phrase, Verb phrase, etc.)
4. Attribute presence feature: This feature is used to check if there is any other attribute present between an attribute and value pair for which relationship is being predicted. Example: 1+ right DP pulse, 2+ left PT pulse In this example, “right DP pulse” attribute is present between 1+ (value) and “left PT pulse” (attribute).
5. Distance feature: This feature captures the distance (number of tokens) between an attribute and value pair. Example: Lactate elevated at 6. In this example, distance between Lactate (attribute) and 6 (value) is 2.

4 Results

This work being an extension of the previously developed system for CLEF eHealth 2014 task, we use the same data which was served as training corpus for CLEF eHealth 2014 task 2. In this work we have experimented only discharge summary records from the available data. We split the total available 136 records into 100 and 36 for training and test purpose respectively.

4.1 Evaluation Criteria

The matching between detected values and true values in actual data are done in a strict manner. These detected values and true values are actually sequences of characters (i.e. string) in a text. Thus, in the strict evaluation, detected and true values are compared literally.

For example, the true representation for attribute in actual data is “blood pressure”; an identified attribute by the system should be identical to it, i.e. “blood pressure” in order to be marked as true. Other outputs including matches in substrings like “blood” or “pressure” will be marked as false.

4.2 Evaluation results

We share the evaluation results of two runs with different features in Table 2. In Run 1 we have used all the features described in section 3.1.2 and first three features described in section 3.2.1. In Run 2 we have used all the features described in sections 3.1.2 and 3.2.1.

Table 2. - Strict evaluation of CRF and SVM

CRF			SVM		
Accuracy Type	Run1 Value	Run2 Value	Accuracy Type	Run1 Value	Run2 Value
Recall	0.93	0.93	Recall	0.75	0.83
Precision	0.97	0.97	Precision	0.92	0.93
F-score	0.95	0.95	F-score	0.83	0.87

4.3 Discussion

The most important finding during this research is that tokenization and feature engineering is very important in NLP based systems.

Another important finding was the distribution of positive and negative training samples. Training data needs to be explicitly checked for the distribution. During the relation extraction testing, accuracy was very low and was not improving. It was found during debugging of the SVM tool that the training data was extremely biased

towards negative data samples. This biasing made the SVM classifier classifying every test data sample to negative category. After the distribution was corrected and biasing was tuned, results were improved drastically.

5 Conclusion

This report describes the approach, algorithms, and tools used in building the numerical attribute and values extraction. CRF algorithm was evaluated and suggested for extracting the attribute and values; it gave 0.95 in F-score. SVM algorithm was evaluated and suggested for relation extraction between an attribute and value, and it gave F-score of 0.87. Though the F-score signifies good accuracy figures, still there is scope of further improvement by designing new features, cross validation, and ensembles of methods (e.g., bagging and AdaBoost). This work can be extended to the problem of non-numerical attribute and value extraction (e.g. dosage mentions such as 'small dose', 'sliding scale', etc. and frequency information like 'two weeks' etc.).

6 References

1. Optimizing Apache cTAKES for Disease/Disorder Template Filling. Nishikant Johri, Yoshiki Niwa, Veera Raghavendra Chikka, http://ceur-ws.org/Vol-1180/CLEF2014wn-eHealth-JohriEt2014.pdf
2. Son Doan and Hua Xu , Recognizing Medication related Entities in Hospital Discharge Summaries using Support Vector Machine, http://dl.acm.org/citation.cfm?id=1944596
3. A System for Identifying Named Entities in Biomedical Text: How Results From Two Evaluations Reflect on Both the System and the Evaluations Shipra Dingare, Malvina Nissim, Jenny Finkel, Christopher Manning, Claire Grover, http://nlp.stanford.edu/jrfinkel/papers/cfg.pdf
4. Two applications of Information Extraction to Biological Science Journal Articles: Enyme Interaction and protein structures. K. Humphreys, G. Demetriou, and R. Gaizuskas, http://citeseerx.ist.psu.edu/viewdoc/similar?doi=10.1.1.40.6118&type=cc
5. I2B2 - Informatics for Integrating Biology and the Bedside, https://www.i2b2.org, Link access date: 05-May-2015
6. CLEF 2014, http://clef2014.clef-initiative.eu/, Link access date: 05-May-2015
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. Louise Rose , Sean P. Clarke, Vital Signs AJN, American Journal of Nursing, May 2010, Volume 110, Issue 5, p 11
9. TEXT2TABLE: Medical Text Summarization System based on Named Entity Recognition and Modality Identification, Eiji ARAMAKI, Yasuhide MIURA, Masatsugu TONOIKE, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.164.9229
10. Stanford named Entity Recognizer, About, Para 2, http://nlp.stanford.edu/software/CRF-NER.shtml
11. Stanford NER Feature Factory, http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html