A New Approach to Extract Meaningful Clinical Information From Medical Notes

Authors: A. Ranjan And R. Bista

Affiliation: Department of Computer Science and Engineering, Kathmandu University, Dhulikhel, Nepal

E-mail - awa.ran@gmail.com and rbista@ku.edu.np

***Abstract*— In medical domain, one of the most important documents is the notes that doctors, nurses or other medical practitioners take during patient interview. These notes contain important information about the patient current condition, symptoms, family history, disease, procedures (like x-ray, lab test etc.), medication and so on. These notes are in plain text English language and are presented in an unstructured way. An unstructured text can be defined as text which contains information not in a common structured format. Since these notes contain very useful clinical information, everyone is willing to extract such information. But lack of defined structure, it makes these texts to be interpreted only by humans and not by any computer program. In structured information, we have only the useful information extracted from the text and other unnecessary information is thrown away. This makes the information to be presented in a definite structure which can be stored in database for further processing. In this approach, we are going to build a medical corpus using the training dataset and apply the corpus in order to extract the information like diagnosis, procedure, drug, habit and vitals from the clinical notes and evaluate it based on different accuracy parameters. We have implemented various Natural Language Processing methods to achieve the results. The system is evaluated based on accuracy, precision, recall and f-score parameters. The experimental result shows that these parameter values are significantly high each being over 0.9.**

Keywords*— *unstructured texts; structured texts; natural language processing; health informatics; corpus; accuracy

I. INTRODUCTION

Natural Language Processing (NLP) [6] is an emerging field of Artificial Intelligence. One of the challenges that we face during NLP implementation is the unstructured content of natural languages. We'll focus on English as Natural Language for this work. Though the English language has definite structures governed by the rules of grammar, in the normal communication, it becomes quite unstructured [20]. Use of abbreviations, ambiguity, not always following the correct grammar rules, incomplete sentences are some of the factors which makes it unstructured. Human brains are smart enough to capture such information and convert into the meaningful information for their use. But it's very hard for the automated computerized system to extract meaningful information from such unstructured texts. Maintaining accuracy is another challenge for such systems. Since we are dealing with medical domain, the accuracy factor is very crucial as it could be life threatening if wrong interpretation is presented.

The main discussion point of this paper is focused on the healthcare and clinical sector where the use of unstructured data [5] is prevalent. There are several instances where the medical practitioners like doctors, pharmacists and nurses generate such unstructured texts while collecting family history, prescribing drugs and so on. These unstructured texts are rich in clinical information and those needs to be extracted as meaningful knowledge for further processing. An unstructured text can be defined as text which contains information not in a common structured format. Whereas, in structured information, we have only the useful information extracted from the text and other unnecessary information are thrown away. The section below discusses about this in detail.

A. Unstructured Text And Its Disadvantages

As discussed above, unstructured texts don't follow any pre-defined structure to store information. It depends on person who is maintaining the note. These can be understood quite easily by humans but not by the computer systems. Here are some of the examples-

- Spoke with pt over the phone. Pt presents with fairly new dx of diabetes, and taking metformin. States this happened about 2 yrs ago and was able to control blood sugars with diet and exercise.
- Pt presents with hyperlipidemia and strong family hx of CAD. Keeps active with job, kids, and softball, and cardio exercise.

As we can see in these notes, there is no regular pattern of information present. Not all notes contain information about diagnosis or drug. Similarly there is not any structured way where the diagnosis comes first, then procedure and then drug. Also these notes contain so many abbreviated texts. Lack of any system to convert that information into structured one makes the medical practitioners' job tedious by needing to read all notes and history each time manually and extracting information from them and again storing it some other place. It consumes a considerable amount of time for them. It’s more tedious when there is any transfer between one departments of hospital to another like from emergency to operation theatre.

B. Structured Information & Its Usefulness

By structured information, it means that the information stored in a regular and general pattern not haphazardly as we saw in previous section example notes. For above notes, the structured way of presenting information would be,

Note 1 -

Diagnosis - Diabetes

Drug - metformin

Habit - Exercise

Vital - blood sugar

Note 2 -

Diagnosis - Hyperlipidemia

Habit - Exercise

Presenting information in the structured way will make it easy to be interpreted by any software program and do further processing like

- Extracting the useful information with some level of accuracy and speed.
- Presenting report based on these facts.
- Suggesting some course of action with some automated suggestion module.

The main objective of this research is as to find out the appropriate method of converting unstructured text to structured information to extract meaningful clinical information from notes entered by medical practitioner using appropriate NLP techniques. Also we’ll be looking into storing the information for future use.

Rest of the paper has been organized as below.

- In section II, we have discussed about the background and some related works in the context of this research.
- In section III, we have discussed about the methodology that has been used for carrying out this work. This section describes all the related steps.
- In section IV, the result and its discussion is presented with the accuracy parameters and their values obtained by this work.
- Section V presents the conclusion and future direction of this work.
- The paper ends with acknowledgement and the list of references.

The domain used for this research purpose is as follows –

- Notes presented in English language
- Notes presented in text file only
- Detection of Diagnosis, Procedure, Habit, Drug and Vitals
- Data from healthcare domain

II. Background and Related Works

In current scenario, there are myriad of research going on in medical NLP sector.

In 2009, Xu *et al.* designed a natural language processing (NLP) based tool MedEX [1] which is used for extracting the medication or drugs information from clinical narratives. It was initially developed using discharge summary. It does well in identifying not only the drug names but other useful information as well like strength, route and frequency etc. A clinical text undergoes three steps in MedEX [1] system to obtain the structured information, 1) Preprocessing in which the sentence boundary detection is done in the clinical texts. 2) Semantic tagging in which each clinical sentence is broken down in tokens and each token is labeled with a semantic category like Drug name, drug strength etc. 3) Parsing then uses context-free grammar to parse the textual sentences into structured form using a chart parser.

G.K. Sovava *et al.* developed system for Mayo Clinical text analysis, cTAKES [2] based on open source NLP for information extraction from Electronic Medical Record (EMR). The system is built on existing open source NLP technologies like the Unstructured Information Management Architecture (UIMA) framework and OpenNLP [15] natural language processing toolkit. This system is trained on clinical domain for creating rich linguistic and semantic annotations. The dataset used for cTAKES [2] is a subset of clinical notes from Mayo Clinic EMR.

In 2010, V. Glara *et al.* developed a system based on cTAKES [2] for the classification of radiology reports that contains the findings leading to suggest that the case is of hepatic decompensation. The system is called The Yale cTAKES [2] extensions for document classification, YTEX [3]. YTEX [3] modified cTAKES [2] by using 1) A regular expression based named entity recognition. 2) Latest version of NegEx algorithm to detect Negation 3) a database module to store the annotations in the database.

Dr. Alan Aronson developed a highly configurable program, MetaMap [4] at the National Library of Medicine (NLM) to map biomedical text to the United Medical Language System (UMLS) [23] Metathesaurus.

III. Methodology

The research methodology used is design and creation method to carry out this work. As per this method, after the background study and awareness of problem, a working system is designed and developed. Data collection is one of the crucial parts of this research. We need to collect the data for training as well as testing purpose. After data collection, data processing is done using the system developed. Once we get the result after this processing, the analysis is done in the result to see how accurate the system was able to classify the data in the appropriate class. Different steps of system development are discussed below.

Since the notes are entered in natural language (English is taken for this research), we are going to use NLP techniques to solve the problem. For the Named Entity Recognition (NER) step, there is the use of corpus based approach. Figure 1 shows a block diagram of the system.

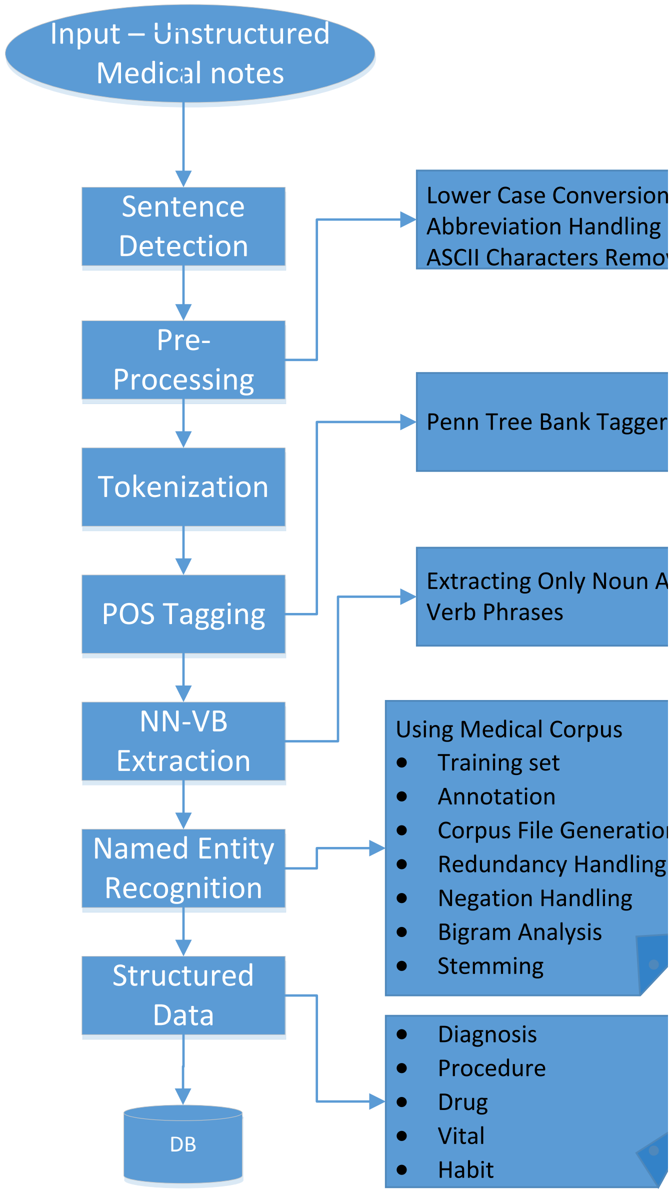


Fig. 1: Block Diagram

The steps are as discussed below.

A. *Sentence Boundary Detection*

In this step, the sentences from clinically rich texts are identified. It detects sentences by using sentence terminators like period (.), question mark (?) etc. E.g.

B. *Preprocessing*

Preprocessing, as the name suggests, is done before actual processing starts and is done in order to standardize the sentences so that it becomes easy in further steps. Following tasks are achieved in this preprocessing phase –

- Abbreviation handling
- Punctuation handling
- Lower case conversion
- ASCII character removal:

C. *Tokenization*

In this step, each sentence is further broken down into individual tokens. A token is an instance of a sequence of characters in some particular document that are grouped together as a useful semantic unit for processing. NLTK [16] tokenize module is used for this. This phase constructs very effective input for next phase which can deal each token wise rather having to deal with large sentences. It can deal with small chunks only.

D. *Parts-Of-Speech Tagging (POS Tagging)*

The next step in the processing is to assign a parts of speech tag to each token. This step is required so that we can limit our search to only those tokens which are tagged as Noun or Verb during further phase of Named Entity Recognition. For POS tagging Penn Treebank tagger [17] is used. This tag-set is developed by The University of Pennsylvania (Penn).

E. *NN-VB Extraction*

Once token is tagged, the main area of concern for further processing would be Noun and Verb phrases. So before the actual recognition of entities like diagnosis, procedure etc., we first extract the Noun and Verb phrases in this phase. This makes recognizer module work on small set of significant data only rather than searching through entire dataset. Only the tokens with following tags are extracted from POS tagger output.

- Noun phrases - NN, NNS, NNP, NNPS
- Verb phrases - VB, VBD, VBG, VBN, VBP, VBZ

F. *Named Entity Recognition (NER)*

This is the phase where we actually recognize one of the following entities from the notes; diagnosis, procedure, drug or medication, vital and habit. In order to effectively implement this module, we have taken the approach of building a medical corpus with training data set. Later on, the corpus is used to find out various entities through the recognition process.

1) *Medical Corpus*

The sole purpose of the corpus building is to apply it in NER. The idea is to generate a rich tagged set of corpus from the available set of data. This corpus will be then used to tag the unstructured text automatically by the system. For the purpose of this research, we need to identify the entities like Diagnosis, Procedure and Drug. So the corpus is focused around these named entities. The corpus is built to recognize one of the following entities as mentioned in the Table I, within any note-

TABLE I. ENTITY TYPES

Entity Type	**Description**	**Example**
Diagnosis	Disease associated with the patient	Diabetes, hypertension, cancer etc.
Procedure	Any procedure done for identification or cure of the disease	MRI, CT Scan, Lab Tests, Therapies etc.
Drug	Medications taken by the patient	Metformin, Lantus, Insulin etc.
Habits	Different habits related to health	Exercise, smoking, jogging etc.
Vitals	Vital signs associated with patient	Weight, height, blood sugar etc.

2) *Method Of Building Corpus*

Following methods were involved in building medical corpus data.

- Training data collection – Corpus is generally associated with some domain. Here we are dealing with medical texts so the domain is limited to medical domain. For this purpose, we collected a large set of texts entered by nurses during patient visits or patient calls with the details of their diagnosis, procedure, medication, vitals and habit.

- Manual annotation - After data collection, the second task is manually annotating the notes to tag the relevant texts. The relevant text is tagged within the span of <START:{type}> (Text) <END> as shown in the Table II below.

TABLE II. ENTITY TYPES

Entity Type	**Annotation Span**
Diagnosis	<START:diagnosis> <END>
Procedure	<START:procedure> <END>
Drug	<START:drug> <END>
Habit	<START:habit> <END>
Vitals	<START:vital> <END>

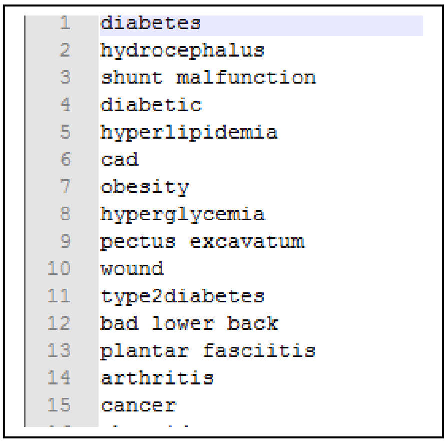
Each text is read manually and put one of these tags for appropriate words with human knowledge as shown in Fig. 2.

- Corpus file generation- After the human annotation is completed; a computer program is built in order to generate the corpus file. Each type of corpus is saved in its separate file name {type}.ner. For example, diagnosis corpus is saved in a file named diagnosis.ner and so on. A sample of diagnosis.ner file is shown in the Fig. 3 below.

- Redundancy handling - While going through such huge amount of data collected, we end up having same elements in the corpus repeated several times in the file bringing redundancy in corpus file. A duplicate removal algorithm is used in order to clean up the redundant information hence bringing enhancement in the recognition performance. Handling redundancy resulted into a higher system performance. The result shows that there is almost 400% gain in the actual processing time with non-redundant corpus.

<START:habit> Smoker <END> > 40 pack
years. *<START:diagnosis> Diabetes <END>*
(<START:vital> HgA1c <END> 7 in 2011),
<START:diagnosis> overweight <END>.
\He\" , has acceptable control of his
<START:diagnosis> diabetes <END> with
<START:vital> HgA1c <END> of 6.3 (was 7 in
2011) & *<START:vital> FBS <END>* 115.
<START:vital> Triglycerides <END> elevated
@ 175, depressed *<START:vital> HDL <END>*
of 29 (huge Risk Factor). *<START:habit>*
Nicotine <END> use high, 1 PPD x 40 years.
NO *<START:habit> exercise <END>* .
<START:diagnosis> Type 2 Diabetes <END> ,
<START:diagnosis> Obesity <END> ,
<START:diagnosis> Gout <END>,
<START:diagnosis> HTN <END>,
<START:diagnosis> Hyperlipidemia <END>

Fig. 2: Manual Tagging

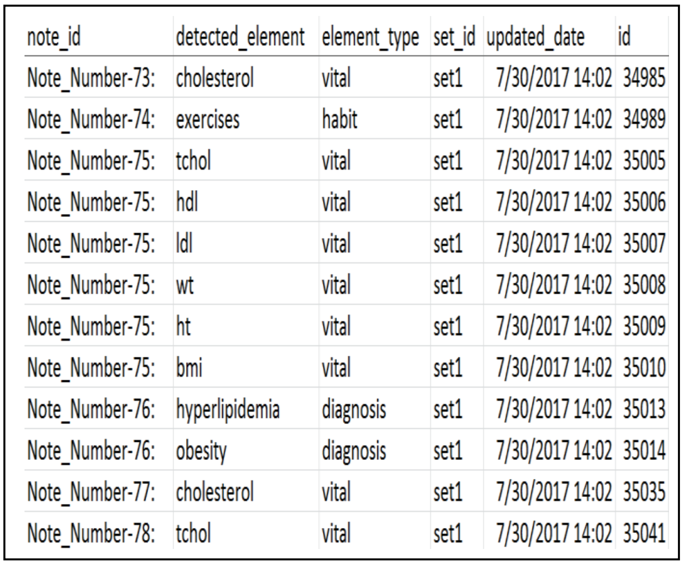


```
diabetes
hydrocephalus
shunt malfunction
diabetic
hyperlipidemia
cad
obesity
hyperglycemia
pectus excavatum
wound
type2diabetes
bad lower back
plantar fasciitis
arthritis
cancer
```

Fig. 3: Snapshot Of Diagnosis Corpus File

G. *Entity Detection*

After the corpus generation, actual entity detection phase is entered using the corpus file. Input to this module is noun-verb extractor output, i.e., Noun and Verb phrases only and the output is recognized entities. Each NN-VB phrase is matched against all corpus files to categorize the element as one of the classes of diagnosis, procedure, habit, vital and drug. Thus identified elements are saved in Database in a structured way as shown in the Fig. 4.



note_id	detected_element	element_type	set_id	updated_date	id
Note_Number-73:	cholesterol	vital	set1	7/30/2017 14:02	34985
Note_Number-74:	exercises	habit	set1	7/30/2017 14:02	34989
Note_Number-75:	tchol	vital	set1	7/30/2017 14:02	35005
Note_Number-75:	hdl	vital	set1	7/30/2017 14:02	35006
Note_Number-75:	ldl	vital	set1	7/30/2017 14:02	35007
Note_Number-75:	wt	vital	set1	7/30/2017 14:02	35008
Note_Number-75:	ht	vital	set1	7/30/2017 14:02	35009
Note_Number-75:	bmi	vital	set1	7/30/2017 14:02	35010
Note_Number-76:	hyperlipidemia	diagnosis	set1	7/30/2017 14:02	35013
Note_Number-76:	obesity	diagnosis	set1	7/30/2017 14:02	35014
Note_Number-77:	cholesterol	vital	set1	7/30/2017 14:02	35035
Note_Number-78:	tchol	vital	set1	7/30/2017 14:02	35041

Fig. 4: Structured Output Saved in Database Table

H. *Negation Handling*

Negation handling is the method of detecting negative elements which will help to reduce false positives in the test result. The negation denoting words are the common words like no, none, free (e.g. smoke free), stops (e.g. stop taking medicine) etc. The negation handling section first segregates all sentences having these negative words. Corpus NER method then used to find out the elements and the elements are tagged as negative elements while saving in the database.

I. *Bigram Analysis*

Bigram analysis is one of the versions of n-gram analysis where we analyze two words together. This technique is very useful in reducing ambiguity and also getting better accuracy. For example, heart attack is a diagnosis but when we analyze single word at a time, we can't detect this as heart and attack will be compared separately and will not be caught as diagnosis. So we need to look at such elements together to get the meaningful output. In bigram analysis, we have first generated bigram corpus out of the corpus files. Then the notes are compared to the bigram corpus file by generating bigram in the notes itself. We have used NLTK bigram feature to implement this.

J. *Stemming*

Stemming is an operation where the stem or root of the word is extracted. For example, the words like smoking, smokes, smoked all have stem as smoke which can be derived using stemming techniques. For this work, we have used NLTK porter stemmer feature. Firstly, we have generated stemmed corpus file which contains the stem of all elements in the corpus. Then the notes under test undergo the stemming process and compared against the stemmed corpus file.

IV. Results analysis and discussion

After the completion of corpus building, we analyzed the accuracy of corpus itself. This was done manually to check if there is any wrong tag element during corpus building. During this phase, we have consulted with physicians also to validate the tagging. Once the corpus set is verified, we were ready to test it in the available dataset. The main focus of the test was to check the extent of the accuracy of this approach so that we can check its effectiveness. We followed an iterative process during testing where the testing was conducted in several pass and the corpus set was enhanced as a feedback of test so that we can cover more during element classification. The structured output is categorized as set1 output for this dataset and the elements which can't be categorized by the system is flagged as "no match xx". The experimental environment used during the test is described in the section below.

A. *Experimental Setup*

For this work, following experimental environment was used.

- RAM - 8 GB
- Processor - Intel(R) Core(TM) i5-3320M CPU @ 2.60GHz, 2 Core(s), 4 Logical Processor(s)
- Operating System - Microsoft Windows 7
- Coding platform - Python 2.7
- Database – MySQL 5.7.9
- Input type – text file

B. *Test Set*

Determining test data is one of the crucial parts of testing. Test data taken is generally some % of the training data that we use to make the corpus. While selecting test data, we need to be careful to take the test data so that it represents the actual real scenario and the result is biased.

We have chosen the simple holdout method [27] where a different ratio of training set and test set is maintained and result was analyzed on those set. We have taken 4 such sets as 95 %-5%, 90% -10%, 85%-15% and 80%-20% of training to test data ratio.

C. *Evaluation Metrics*

For evaluating the results, we used the standard metrics namely accuracy, precision, recall and F-Score. The metrics formula [2] of these performance evaluation metrics are given as below.

$$accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + FalsePositives + TrueNegatives + FalseNegatives} \tag{1}$$

$$precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{2}$$

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{3}$$

$$F\text{-}Score = \frac{2 * precision * recall}{precision + recall} \tag{4}$$

Where,

True Positive (TP) is the condition when the system is accurately able to identify the elements with correct type. So if the element is correctly classified as one of the diagnosis, procedure, habit, vital and drug, it falls under true positive.

True Negative (TN) is the condition when accurately detects that the element doesn't belong to any category. In our case, if the element is flagged as "no match xx" correctly, the count of such values is considered as True Negative values.

False Positive (FP) is the condition when the system flags the element to be classified as one of the diagnosis, procedure, habit, vital and drug, but actually the element is should not have been detected. Main reason for false positive is the ambiguity and negativity due to which the system thinks that it got a match but in real context that is not true.

False Negative (FN) is the condition when the element is not detected by the system but should have been detected in real case. So in our case, if the element is flagged as "no match xx" but in real case, it falls in one of the groups - diagnosis, procedure, habit, vital and drug; then this contributes to false negative.

D. Result Analysis

The result analysis is performed on two ways as follows –

1. Result analysis based on varying training data size – In this analysis process, we have taken different 4 range of training data containing 1000, 5000, 10000 and 15000 sentences. We then checked the number of corpus elements and accuracy on all these training set to determine the trend and to confirm what size of training data will be more suitable.

   Fig. 5 shows the trending graph of number corpus elements detected while taking various training size.

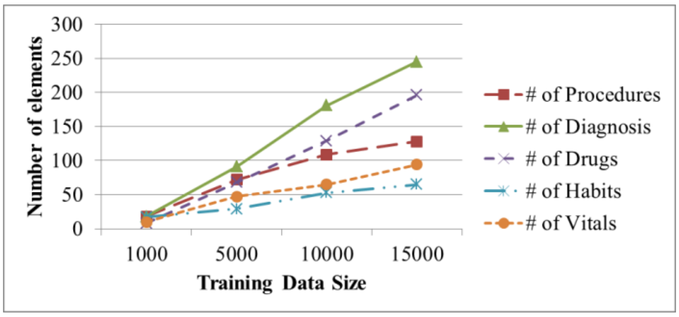


Fig. 5: Training Data Size vs. Corpus Elements

   Fig 6 below shows the trend of accuracy parameters while taking various training data size.

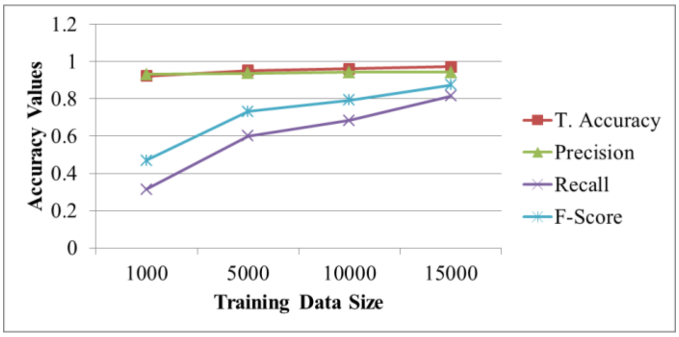


Fig. 6: Training Data Size vs. Accuracy

   Here T. Accuracy denotes Total Accuracy which is same as specified in equation (1).

   From these two graphs, we can conclude that increasing training data set gives better results both in terms of making the corpus size richer and making the accuracy better of the system. For the rest of the analysis and testing, we have taken the corpus generated by the training set of 15000 sentences.

2. Result analysis based on varying test set – In this analysis, we have chosen 4 different sets of test data as mentioned in Test Data section above. For this analysis, we have taken the corpus file generated using 15000 training data. Fig 7 shows the accuracy trend on varying test data.

3.

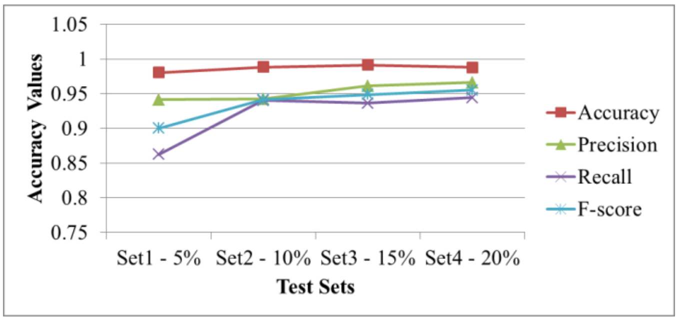


Fig. 7: Test Set vs Accuracy values

As we can see from this trending graph, the accuracy parameter values either increases or remains same across these test sets.

Table III shows the overall values of the accuracy parameters which we were able to achieve by using corpus based NER approach.

TABLE III. EVALUATION METRICS

Evaluation Metrics	**Value**
Accuracy	0.986
Precision	0.953
Recall	0.917
F-Score	0.934

.

The system developed is combined with three additional features i.e. negation handling, bigram analysis and stemming. These three features add greatly to such high values of accuracy parameters.

V. CONCLUSION AND FUTURE WORK

Extraction of meaningful information clinical notes is really challenging task as it needs to be very accurate. The system needs to present the information accurately hence the accuracy parameters should be of high value. The accuracy of the system discussed in the paper is in higher side but we can achieve more accuracy by increasing more number of sentences in the corpus.

Some of the limitations of this work include; being limited to English language only, input format supported is text files only and the elements detected from notes are limited to Diagnosis, Procedure, Drugs, Habits and Vitals or Labs only.

We can introduce more N-gram analysis (like Trigram) to have more accuracy and reducing more ambiguity. Corpus size can be increased as much as possible to bring more coverage in the Named Entity Recognition process.

Currently this work is limited to detecting the lab elements only but it can be extended to detect the lab values too from the notes. This work can also be extended in order to build a learning system which will allow to add the undetected elements from this work but which are actually the structured data, in the corpus. We can also extend this work to get the data from speech or images (scanned copies of notes) and then do the analysis as per based on this work.

ACKNOWLEDGEMENT

We would like to thank Deerwalk Inc. for providing the clinical data needed for the research as well as supporting financially for this research work. Also we would like to extend our gratitude to Lekhnath Bhusal for the guidance during initial system design. We would also like to thank Dr. Manoj Choudhary for helping with defining correct classification during clinical corpus building phase. Our appreciation goes to Subigya Nepal for evaluating the existing tools like MedEX.

REFERENCES

[1] H. Xu, S.P. Stenner, S. Doan, K.B. Johnson, L.R. Waitman and J.C. Denny, “MedEx: a medication information extraction system for clinical narratives”, Journal of the American Medical Informatics Association (JAMIA), pp. 19-24, October 2009

[2] 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 (JAMIA), pp. 507-513, June 2010

[3] V. Garla, V.L. Re III, Z. Dorey-Stein, F. Kidwai, M. Scotch, J. Womack, A. Justice and C. Brandt, "The Yale cTAKES extensions for document classification: architecture and application”, Journal of the American Medical Informatics Association (JAMIA), pp. 1-7, April 2011

[4] A.R. Aronson, “MetaMap: Mapping Text to the UMLS Metathesaurus”, National Library of Medicine, July 2006

[5] M. Stonova, “Unstructured Data in Healthcare”, IJBH, 2014

[6] E.D. Liddy, "Natural Language Processing”, Encyclopedia of Library and Information Science 2nd Edition, 2001

[7] C. Ronan, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural Language Processing (Almost) from Scratch", Journal of Machine Learning Research 12, pp. 2493-2537, August 2011

[8] Y. Wu, J.C. Denny, S.T. Rosenbloom, R.A. Miller, D.A. Giuse, Dr.Ing and H. Xu, "A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries", Department of Biomedical Informatics, Department of Medicine, School of Medicine, Vanderbilt University, Nashville, TN

[9] O. Bodenreider, J. Willis and W. Hole, "The Unified Medical Language System", National Library of Medicine, 2004

[10] D. Jurafsky and J.H. Martin, "Speech and Language Processing", Pearson Education, Second Edition, 2014

[11] S. Pal, "Language Model to detect Medical Sentences using NLTK", URL - http://sujitpal.blogspot.com/2013/04/language-model-to-detect-medical.html , April 2013 (visited on August 2015)

[12] J. Perkins, "Python Text Processing with NLTK 2.0: Creating Custom Corpora", URL- https://www.packtpub.com/books/content/python-text-processing-nltk-20-creating-custom-corpora, November 2010 (visited on August 2015)

[13] A. Coffman and N. Wharton, "Clinical Natural Language Processing Auto-Assigning ICD-9 Codes", 2007

[14] O. Bodenreider, J. Willis, and W. Hole, "The Unified Medical Language System", National Library of Medicine, 2004

[15] D.R. Radev, "Introduction to Natural Language Processing", Coursera, University of Michigan, December 2016

[16] OpenNLP, URL - https://opennlp.apache.org/ (visited on December 2014)

[17] NLTK, URL - http://www.nltk.org/book/ (visited on August 2015)

[18] B. Santorini, "Part-of-speech tagging guidelines for the Penn Treebank Project. Technical report MS-CIS-90-47", Department of Computer and Information Science, University of Pennsylvania, 1990

[19] R. Wolniewicz, "Auto-Coding and Natural Language Processing", 3M Health Information Systems

[20] C. Trim, “Natural Language Understanding of Unstructured Data”, URL- https://www.ibm.com/developerworks/community/blogs/nlp/entry/natural_language_understanding_of_unstructured_data1?lang=en, IBM

[21] Test Statistices, URL - http://groups.bme.gatech.edu/groups/biml/resources/useful_documents/Test_Statistics.pdf (visited on August 2017)

[22] Rates, URL - http://www.uta.fi/sis/tie/tl/index/Rates.pdf (visited on June 2017)

[23] O. Bodenreider, J. Willis and H. William, “The Unified Medical Language System - What is it and how to use it?”, National Library of Medicine, 2004

[24] Deerwalk Inc., URL – http://www.deerwalk.com

[25] I. Guyon, “A scaling law for the validation-set training-set size ratio”, AT&T Bell Laboratories, 1997

[26] A. Clark, C. Fox and S. Lappin, “The Handbook of Computation Linguistics and Natural Language Processing”, Wiley-Blackwell

[27] Test Set, URL - https://en.wikipedia.org/wiki/Test_set (visited on August 23, 2017)

[28] Z. Reitermanov´a, “Data Splitting”, Charles University, Czech Republic, 2010

[29] G.C.Garbacea, E.Tsagkias and M. deRijke, “Feature Selection and Data Sampling Methods for Learning Reputation Dimensions”, The University of Amsterdam at RepLab, 2014

[30] N. Madnani, "Getting Started on Natural Language Processing with Python"