***A New Approach to Extract Meaningful Clinical Information From Medical Notes***

*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.

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

# Introduction

Natural Language Processing (NLP) 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. 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 texts 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.

**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.

**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 mentioned above will make it easy to be interpreted by any software program and do further processing

# 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) Metathesaurus.

# Objectives

Object of this research is as follows -

* To find out the appropriate method of converting unstructured text to structured information
* To extract meaningful clinical information from notes entered by medical practitioner
* To store the information for future use
* To implement the appropriate NLP technique to solve the problem.

# methodology

## 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. The steps are as discussed below.

## 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.

* Input note – [“FBS & hgA1c both slightly improved, but still prediabetes (HgA1c = 5.8%). But did instruct on diet/exercise.”]
* Output sentences - [“FBS & hgA1c both slightly improved, but still prediabetes (HgA1c = 5.8%).”, “But did instruct on diet/exercise.”]

## 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: Abbreviated texts are replaced by its full form. A list based replacement approach is used for this. E.g. pt is expanded as patient; dx is expanded as diagnosis etc.
* Punctuation handling: The texts having punctuation and denoting negation, like “don’t”, “hasn’t” etc. are converted into actual negative form like “do not”, “has not”. This makes the negation handling part easy to detect the negative scenarios.
* Lower case conversion: In order to bring standardization and reduce the case conversion effort during named entity detection phase, all texts are converted in lower case characters.
* ASCII character removal: Since the notes are maintained in different systems, there is chance of having different ASCII characters which makes the program to fail. So ASCII characters are removed before further processing.

## 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. Following is an example of tokenizer –

* Input sentences - [“FBS & hgA1c both slightly improved, but still prediabetes (HgA1c = 5.8%).”, “But did instruct on diet/exercise.”]
* Output tokens - ['FBS', '&' , 'hgA1c', 'both', 'slightly', 'improved', ',', 'but', 'still', 'prediabetes', '(', 'HgA1c', '=', '5.8', '%', ')', '.', 'But', 'did', 'instruct', 'on', 'diet/exercise', '.' ]

## 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). The example of tagging using this tagger is as follows –

* Input tokens - ['FBS', '&' , 'hgA1c', 'both', 'slightly', 'improved', ',', 'but', 'still', 'prediabetes', '(', 'HgA1c', '=', '5.8', '%', ')', '.', 'But', 'did', 'instruct', 'on', 'diet/exercise', '.' ]
* Output POS tags – [('FBS', 'NNS') ('&', 'CC') ('hgA1c', 'NNP') ('both', 'DT') ('slightly', 'RB') ('improved', 'VBN') (',', ',') ('but', 'CC') ('still', 'RB') ('prediabetes', 'VBZ') ('(', ':') ('HgA1c', 'NNP') ('=', ':') ('5.8', 'CD') ('%', 'NN') (')', ':') ('.', '.') ('But', 'CC') ('did', 'VBD') ('instruct', 'NN') ('on', 'IN') ('diet/exercise', 'JJ') ('.', '.')]

## 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

## 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.

* **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-

1. 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. |

* **Method of building corpus**

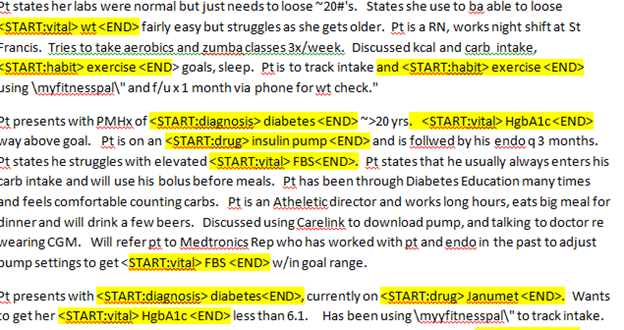
Following methods were involved in building medical corpus data.

* 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. Around 5,000 such sentences are used for building the corpus.
* 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.

1. entity types

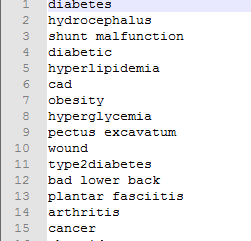
|  |  |
| --- | --- |
| **Entity Type** | **Annotation Span** |
| Diagnosis | <START:diagnosis> <END> |
| Procedure | <START:procedure> <END> |
| Drug | <START:drug> <END> |
| Habit | <START:habit> <END> |
| Vitals | <START:vital> <END> |

Each text is read manually and put one of these tags for appropriate words with human knowledge as shown in Fig.1 below –



1. Manual Tagging

* 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. 2 below.

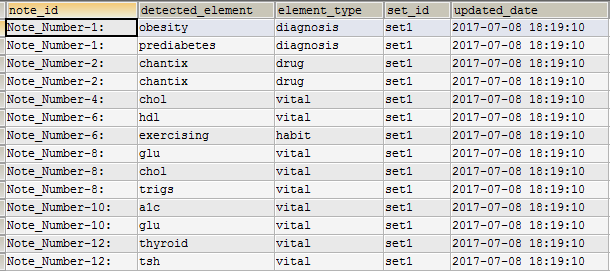


1. Snapshot Of Diagnosis Corpus File

* 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.

## 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 output is recognized entities. Thus identified elements are saved in Database in a structured way as shown in the Fig. 3 below –



1. Structured Output Saved in Database Table

# Results analysis and discussion

**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

**Test set:**

Test data taken was around 25% of the training set. The test data was an unseen data which went through the corpus based recognition system and detected the entities.

**Evaluation metrics:**

For evaluating the results, we used the standard metrics as specified below –









Where,

True Positive is the condition when the system is accurately able to identify the elements with correct type.

True Negative is the condition when accurately detects that the element doesn’t belong to any category.

False Positive is the condition when the system flags the element to be detected but actually the element is should not have been detected.

False Negative is the condition when the element is not detected by the system but should have been detected in real case.

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

1. evaluation metrics

|  |  |
| --- | --- |
| Accuracy | 0.97 |
| Precision | 0.95 |
| Recall | 0.73 |
| F-Score | 0.82 |

# The values can be further be refined by taking more corpus size. Currently we take only 5000 sentenced corpus. The more we increase the corpus size, the more NER elements can be detected contributing to the high true positive count. In the same time, it is also a risk of increasing false positive detection too.

# application areas

The system having the capability of automatically converting unstructured text to structured knowledge has huge implications in the medical sector. Some of them are discussed here briefly,

* Extracting information from family
* Efficiently extracting information from clinical notes
* Extracting information from prescription
* Extracting information from discharge
* Machine learning and prediction systems in healthcare
* Reporting systems in Health sector
* Developing standards

# limitations and future work

**Limitations:**

1. The research work is limited to English language only. The notes taken in English language can only be analyzed using this.
2. The input format supported is text files only.
3. The resulting system of this research is limited to work on single platform, not distributed environment support.
4. Analysis on only Noun and Verb phrases. It doesn’t include other Parts-of-speech like adjective, preposition etc.
5. Corpus size is limited to 5000 sentences.
6. The elements detects from notes is limited to Diagnosis, Procedure, Drugs, Habits and Vitals or Labs only.

**Future work:**

1. This work can be extended to include morphological analysis so that once we build the corpus; it can automatically detect various form of element like gerund, past participle etc. E.g., if we have “run” as valid element in the corpus tagged as habit, it should automatically detect its variations like “runs”, “running” etc. as habit. This will reduce the morphological redundancy as well.
2. N-gram analysis (like Bigram, Trigram etc.) can be included to have more accuracy and reducing more ambiguity.
3. Corpus size can be increased as much as possible to bring more coverage in the Named Entity Recognition process.
4. Currently the work is limited to detecting the lab elements only but it can be extended to detect the lab values too from the notes.
5. 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.
6. We can extend this work to get the data from speech or images (scanned copies of notes) and then do the analysis as per this work.

##### 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. E.D. Liddy, "Natural Language Processing”, Encyclopedia of Library and Information Science 2nd Edition, 2001
6. 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
7. 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
8. O. Bodenreider, J. Willis and W. Hole, "The Unified Medical Language System", National Library of Medicine, 2004
9. D. Jurafsky and J.H. Martin, "Speech and Language Processing", Pearson Education, Second Edition, 2014
10. 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)
11. 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)
12. A. Coffman and N. Wharton, "Clinical Natural Language Processing Auto-Assigning ICD-9 Codes", 2007
13. O. Bodenreider, J. Willis, and W. Hole, "The Unified Medical Language System", National Library of Medicine, 2004
14. D.R. Radev, "Introduction to Natural Language Processing", Coursera, University of Michigan, December 2016
15. OpenNLP, URL - https://opennlp.apache.org/ (visited on December 2014)
16. NLTK, URL - http://www.nltk.org/book/ (visited on August 2015)
17. 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
18. R. Wolniewicz, "Auto-Coding and Natural Language Processing", 3M Health Information Systems
19. Test Statistices, URL - <http://groups.bme.gatech.edu/groups/biml/resources/useful_documents/Test_Statistics.pdf> (visited on August 2017)
20. Rates, URL - <http://www.uta.fi/sis/tie/tl/index/Rates.pdf> (visited on June 2017)
21. N. Madnani, "Getting Started on Natural Language Processing with Python"