CLINICALLY SIGNIFICANT INFORMATION EXTRACTION FROM RADIOLOGY REPORTS

by

Nidhin Nandhakumar

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
July 2017

To my Mom(Suma Nandhakumar) Dad(Nandhakumar K) and to all my Friends and Family who have stood behind me for the past years

Table of Contents

List of Tables

List of Figures

Abstract

Radiology reports are one of the most important medical documents that a diagnostician looks into, especially in the emergency situations. They provide the emergency physicians with critical information regarding the condition of the patient and help the physicians take immediate action on urgent conditions. However, the reports are complex and unstructured.

We developed a machine learning system to efficiently extract the clinically significant parts and their level of importance in radiology reports. The system also classifies the overall report into critical or non-critical which help radiologists in identifying potential high priority reports. As a starting point, the system uses Chest X-RAY reports of adults (de-identified) and provides the doctors with 3 levels of medical phrases namely high-critical conditions, critical conditions and non-critical conditions. We used Conditional Random Field (CRF) to identify clinically significant phrases with an average F1-score of 0.75.

The CRF Model is used as a filter with the web interface which highlights the medical phrases and their criticality level to the emergency physician. The overall classification of the report is identified using Stochastic Gradient Descent and features used are phrases extracted from the CRF model which provides an average accuracy of 0.85.

List of Abbreviations Used

CMM Conditional Markov Model

CRF Conditional Random Field

EHR Electronic Health Record

L-BFGS Limited memory BFGS

MEMM Hidden Markov Model

MMTx Meta Map Transfer

NER Named Entity Recognizer

NLP Natural Language Processing

POS Part Of Speech

RF Random Forest

SGD Stochastic Gradient Descent

SVM Support Vector Machine

UMLS Unified Medical Language System

Acknowledgements

I would like to extend my sincere gratitude and appreciation to Dr. Evangelos Milios for his constant guidance, support and encouragement. It has been a pleasure to work under your aegis. I would like to thank him for giving me this incredible opportunity to learn and pursue my passion on text analysis and machine learning. I would also like to thank Ehsan Sherkat who have helped to steer in the right direction with his experience. He has been a wonderful colleague and friend who have given me the confidence and support thoughout this study. I am also thankful to Dr. Hong Gu for her support and guidance in developing this system.

I would also like to thank Michael Butler and Jessie Kang for their incredible support during the period of this research. They have devoted time from their busy schedule to provide us with annotations which are used for training the machine learning model. I would also like to thank Lisa Ling for her support during this research work.

With a graduate research funding from NSERC Engage Grant with Palomino System Innovations, it helped me to stay focused on my research work and the schedule. I would like to express my gratitude to NSERC committee for providing a grant for this work.

I would also like to thank my Friends and Family who have kept me motivated and supported me during my research in Dalhousie University.

Chapter 1

Introduction

Information Extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents [1]. Information extraction has been one of the most fascinating and challenging areas of natural language processing. The most important aspect of information extraction is to extract 'key' information from the given free form text. However, identifying 'key' information is a complex task due to a number of factors. One such complexity is analyzing the context of the information in a given sentence. Usually, depending on the context (what happened in previous words and what is happening in next words of the sentence), the information extracted could be relevant or irrelevant. For example. If the system is trying to extract 'names' of people from a given e-mail, the word such as 'Ltd' can distinguish between a company name and a human name if this word comes after the observed word. If the algorithm does not consider this context, the extracted name could be a person's name instead of a company name. Information extraction plays a crucial role in a medical domain where patient information is usually stored as text information. Up until recent times, these free-form texts were not used for future diagnostic analysis or patient profiling. One of the main reasons is the difficulty involved in extracting information from medical documents. Medical reports contain more medical terms which are highly domain oriented and normal methods of information extraction would perform poorly.

One of the key resources of information used by doctors (especially emergency physicians) are radiology reports of the patients (Hall, 2000). The radiology report comprises key medical observations dictated by the radiologist when analyzing the patient's medical imaging reports (for example, x-rays) and these are automatically transcribed to text. It is the emergency physician who makes the decision on the

[1] https://en.wikipedia.org/wiki/Information_extraction

treatment of the medical conditions. In the case of long radiology reports, the doctor may miss some of the key observations made by the radiologist. Another complexity in the processing of radiology reports is the presence of transcription errors. A sample of chest x-ray radiology report is shown in Figure 1.1

CHEST X-RAY AND LEFT ELBOW absent . CLINICAL
INDICATION:
a year old who fell . Now pain in the left elbow .
FINDINGS:
Severe degenerative changes noted within the left elbow consisting of osteophytosis of the olecranon and coronoid process of the ulna . as well there are multiple ossific densities projecting within the anticipated location of the anterior elbow joint which may represent intraarticular bodies . there is no fracture of the ulna or radius . no effusion is identified . there is significant subchondral sclerosis osteophytosis and narrowing at the distal radial ulnar joint . As well significant degenerative change at the first metacarpal carpal articulation consisting of joint space narrowing and subchondral sclerosis . Incidental note is made of extensive vascular calcification .
IMPRESSION:
No obvious underlying elbow fracture . there are extensive degenerative changes within the elbow joint as described above . If there is ongoing clinical concern for a fracture a CT of the elbow is recommended for further evaluation .

Figure 1.1: Example of Chest X-Ray radiology report of a patient

The main purpose of this research was to change the unstructured data obtained from radiology reports into structured information so that this information can be further used in machine learning for diagnoses. The structured information also provides a template for future structure design for the radiologists to directly record the information into a structured data format. The key information extracted from the reports are the critical level of the medical conditions of the patient. This is a complex task because the radiology reports contain lot of information even on small reports and identifying medical terms or conditions and identifying the critical level of them require significant domain knowledge even for a human.

Our proposed model tries to aid the emergency physicians in automatically identifying key medical observations from the radiology report, based on the criticality level of the medical phrases, using machine learning and a web-based visual interface. The system highlights the medical phrases on the fly, based on their criticality values for the doctors. We also classify the overall report to identify if the patient is in need

of urgent treatment. We have listed our main contributions as the following:

- The design of a novel system which identifies the medical phrases and their associated criticality values and presents this information in a visual interface. In terms of performance, our proposed method is able to achieve similar performance to human annotators when identifying key phrases and their criticality level.

- The design of a Web-based tagging system which can be used by doctors for annotating the radiology reports to provide training data for the machine learning model.

- We have also managed to improve the accuracy of word segmentation and spelling correction algorithms and have tuned them for use in radiology reports.

- Designing a novel binary classification system for extracting radiology reports of critical-condition patients. The proposed approach was able to achieve better performance as compared to using 'bag of words' having tf-idf weights. We have managed to use a novel list of features for better classification of the radiology reports.

We start this thesis with an overview of related models or systems which use radiology or similar medical reports for extracting information from unstructured data. We present the overall flow of our proposed model for extracting information from the radiology report in Chapter 3. The implementation details of each of the modules are mentioned in subsections. We then compare and evaluate the performance of our system in Chapter 4. Finally, we conclude the thesis in Chapter 5.

Chapter 2

Related Work

There have been a lot of work done on analyzing and extracting information from medical documents. The medical field is becoming more and more aware of the advantages and importance of using advanced machine learning[1] and natural language processing[2] methods for analyzing the text and unstructured medical records. In this thesis, we try to analyze the previous work done by various researchers on analysis of medical text documents and information extraction from them using various methods. We divided the previous work section into two sub-sections. First, we discuss various research works done on extracting specific information from medical reports such as diagnostic of a medical condition or finding specific medical dosage information. On the second section specifically, discuss on the information extraction from radiology reports.

2.1 Information Extraction from Medical reports

Extracting information from medical reports are a complex task due to the unstructured format of the reports. Even though there is a template form for some of the reports, the template could change as per institutions and standards. Because of these reasons, it is difficult to extract multiple types of information from a single report. Because of this, many of the information extraction models try to extract one specific information from the patient records such as dosage information or diagnosis of a specific type of disease for the patient. Some researchers use machine learning techniques alone for extracting the information required while some, use a combination of machine learning and rule-based models for extracting the information.

[1] https://en.wikipedia.org/wiki/Machine_learning
[2] https://en.wikipedia.org/wiki/Natural_language_processing

One research which tries to extract information from medical records is the work done by Meystre et al. (2010). This research was focused on extracting medication information from the patient's EHR files. They extracted 5 separate classes as Dosage, Route, Frequency, Duration and Reason. They implemented a hybrid system which consisted of machine learning and pattern matching techniques for extracting the medication information.One of the main resource used is the MMTx which is a java version of meta map[3] which can map the terms with the UMLS concepts (Bodenreider, 2004) . The model consisted of the main class which is the medication and several subclasses which can help to predict the medication class. Two slots are also used along with the main class. The first slot tells if the entity is part of the list or narrative list and second slot which tries to link the annotated text with the subclasses.The model consisted of a document structure analysis module which identifies the structure of the document based on the pattern matching and regular expression so that the patient report can be segmented into various sections and can be easily processed for identifying the subclasses. Some sections can be eliminated if not useful for the given task. Sentence detection is done based on regular expression and pipelined to POS tagger. MMTx is used to identify the UMLS concepts for the medications and to identify possible reasons for the prescription of the medication. The MMTx concept is recognized for each sentence.Context analyzer uses the context tool to identify the context of the text. Medical terms are extracted based on regular expression. The medication reconciliation is also based on regular expressions which look for several patterns which can identify the subclass sections. the class and sub-class model is given in Figure 2.1

Another similar system Patrick and Li (2009) uses similar techniques as (Meystre et al., 2010). The model is also used to extract medication information from the patient discharge file. The information to be extracted are dosage, mode, frequency, duration, reason, and context. The train data consisted of 130 reports annotated by the physician and further reference by the researcher. The test set consisted of 30 reports. The model consisted of the following steps. This system uses CRF to identify the entities, and build pairs for each medication relationship (only consider drug

[3]https://metamap.nlm.nih.gov/

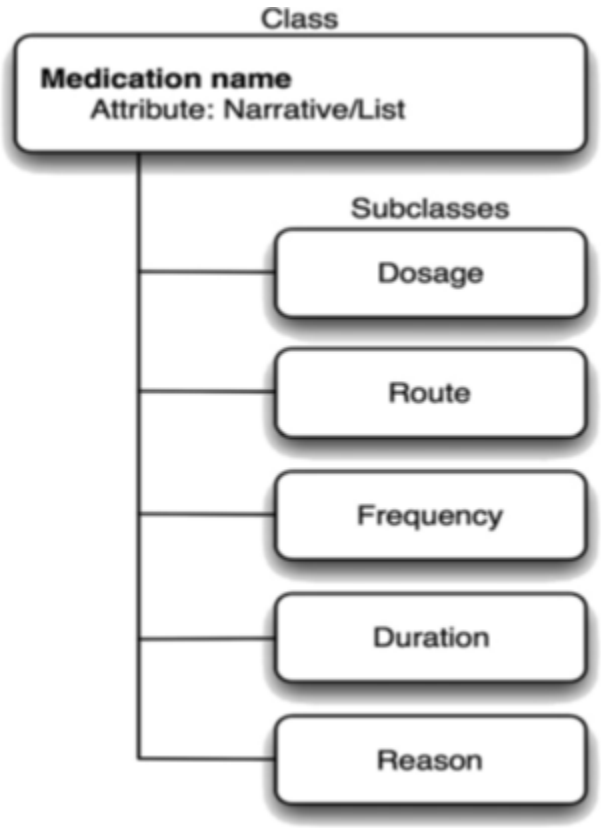

Figure 2.1: Class model for textractor system adopted from Meystre et al. (2010)

and its related entity, since the whole related entities, such as dosage, frequency, etc., could be further connected based on the drug). The output is then classified by the binary classifier SVM. The final medication entries are based on the results from the CRF and SVM. Initially, the records are divided into sentences and POS tags were identified. Seven features from the patient records are identified as drug, dosage, mode, frequency, duration, reason and other morphological features. The features are then trained on the CRF model. Finally, the output of CRF is converted into SVM input. The SVM is then used to classify the relationship between medication pairs. The context identification unit identifies which context the entry is related to. The hybrid design model is shown in Figure 2.2

Some other interesting works include the use of the Decision Trees to identify the clinical findings and recommendations in the radiology report by Dreyer (2014). However, the exact implementation details of this model are not provided by the author

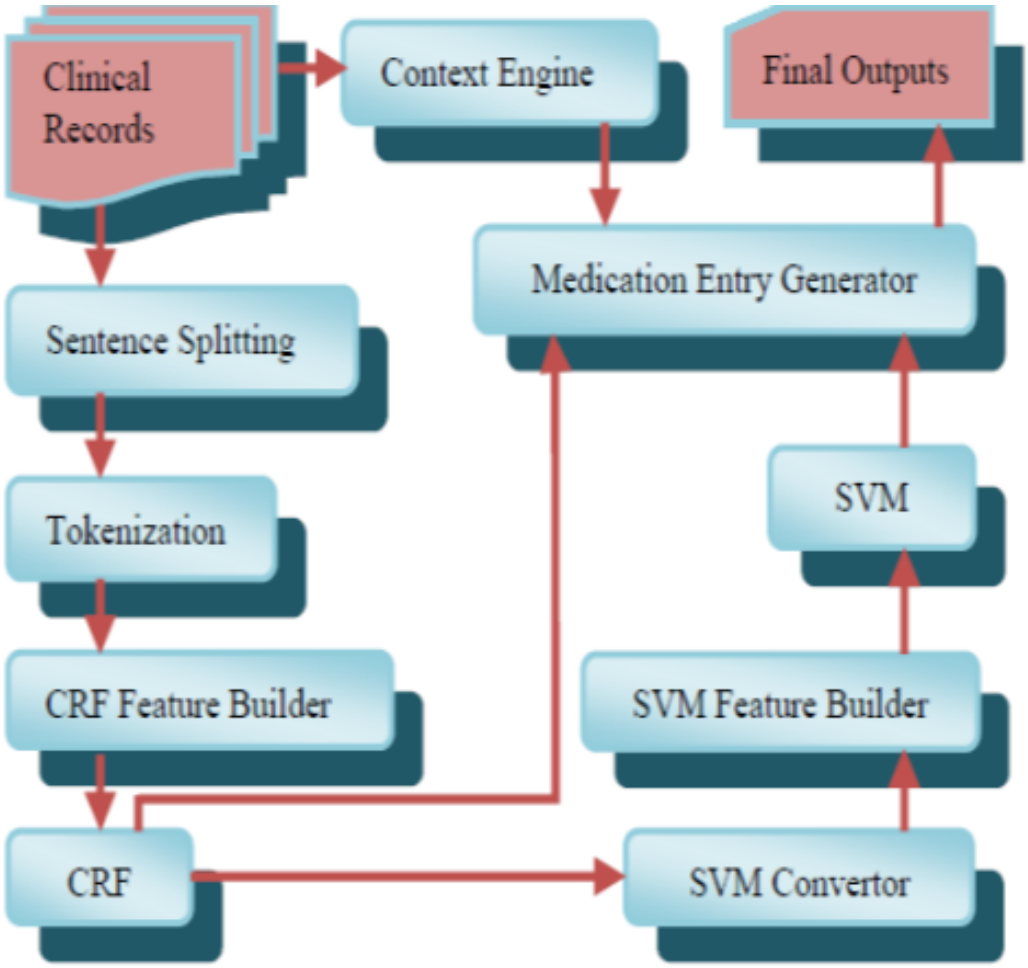

Figure 2.2: Hybrid System Model for (Patrick and Li, 2009)

for replicating the results. Another system that uses patient's discharge summary report for extracting medication information which includes drug names and dosage information is Xu et al. (2010). This model uses regular expressions, dictionary looks up to identify the medication and dosage information from the file. The main part of this system is semantic, which uses pre-loaded lexicons to initially tag the parts of a sentence into various types such as drug name, modifier etc. Then a second stage parsing is done to further tag the uncertain values to get final tagging. The model for Medley system is shown in Figure 2.3

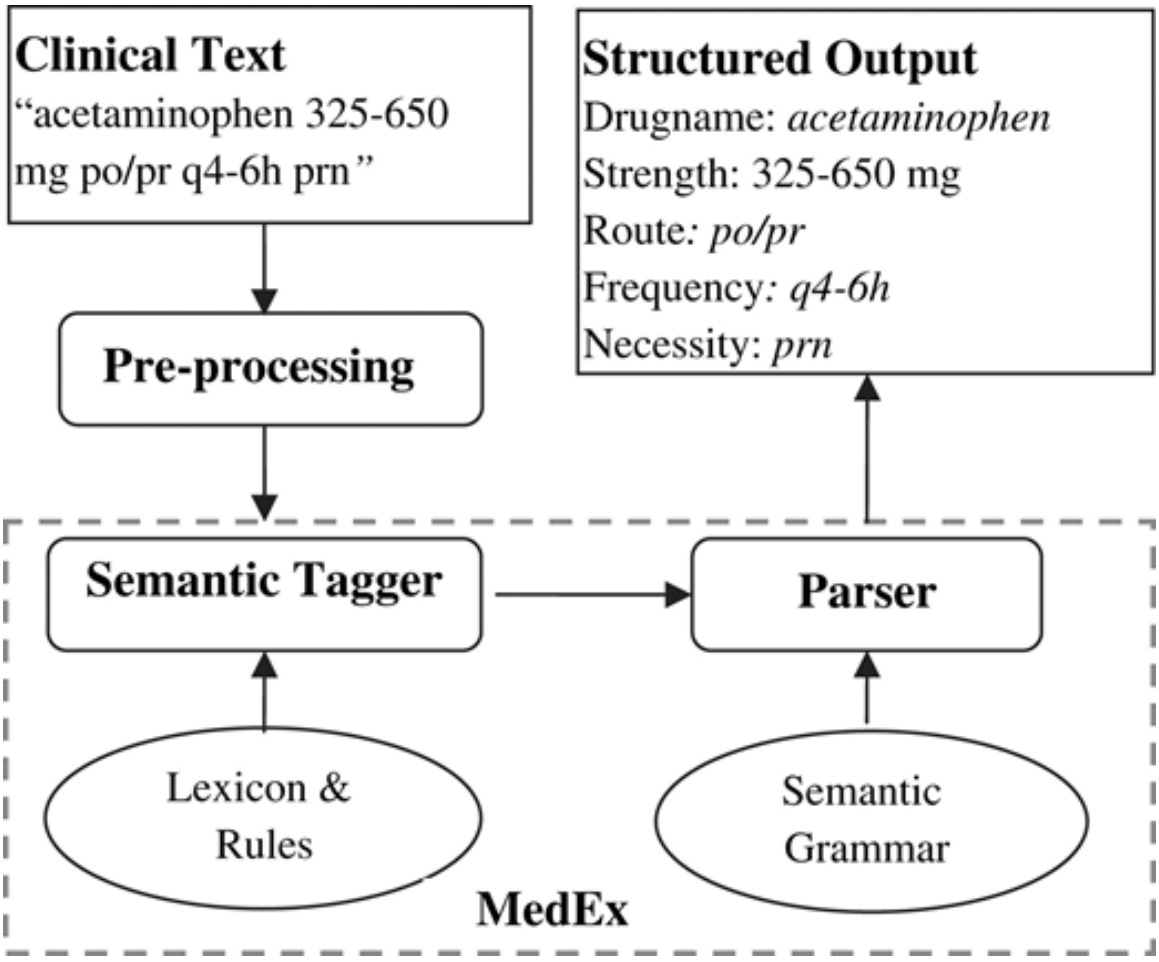

Figure 2.3: System model for Medex adopted from Xu et al. (2010)

2.2 Information extraction from radiology reports

Radiology reports are a special case of medical reports. They do have a vague structure which can change depending on the reports and also based on the radiologist. They also have multiple information in them. Because of this reason, most of the research work done tries to identify a specific information from the radiology reports. Another issue with radiology reports is the privacy of the reports. usually, the reports are de-identified which removes any personal information related to the patient including age and sex. However, in practice, these information plays a crucial role in diagnosis. A medical condition identified for a 24-year-old male could be benign while malignant for a 60-year-old female. In this section, we identify several studies which try to extract information from radiology reports.

One such study Yetisgen-Yildiz et al. (2013) uses a text processing pipeline to extract recommendations from the radiology reports. CTakes uses a combination of rule-based and machine learning methods to extract clinically significant information from the radiology reports. The model is an open source system which includes several components (Sentence boundary detection, tokenizer[4], normalizer[5], POS tagger [6], Shallow parser, and NER[7] including status and negation annotator). The tokenizer will detect the various tokens including the measurement, person, title etc.The normalizer normalizes the word based on various properties of the word including the case, punctuation, generative markers etc. The NER is implemented based on a dictionary look up for noun phrases, and the negation detector detects negation words associated with the named entities.The various semantic attributes identified are: (1) the text span associated with the named entity (span attribute), (2) the terminology ontology code the named entity maps to (concept attribute), (3) whether the named entity is negated(negation attribute), and (4) the status associated with the named entity with a value of current, history of, family history of, possible(status attribute). The model validation is done by 10 fold cross validation. The system needs a rich up to date dictionary for producing accurate results and the system does not do well at a complex level of synonym. The system design is given in Figure 2.4

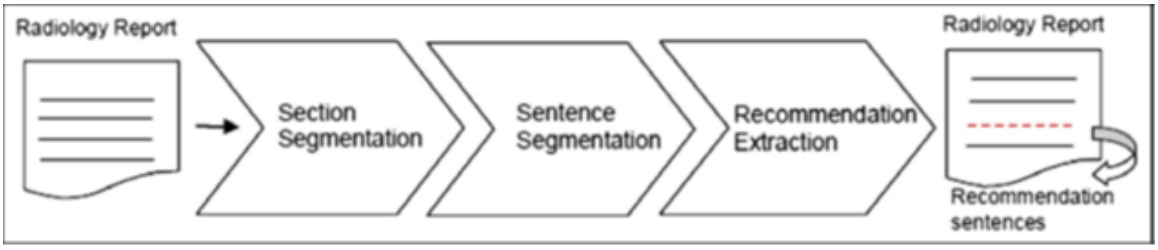

Figure 2.4: System Model for Yetisgen-Yildiz et al. (2013)

Some of the recent work includes the extraction of tumor information from radiology reports (Yim et al., 2016). In this model, the objective was to extract tumor

[4]https://en.wikipedia.org/wiki/Tokenization_(lexical_analysis)
[5]https://en.wikipedia.org/wiki/Text_normalization
[6]https://en.wikipedia.org/wiki/Part-of-speech_tagging
[7]https://en.wikipedia.org/wiki/Named-entity_recognition

information for Hepato Cellular Carcinoma[8] disease. They used patients Electronic Medical Record for identifying information such as tumor number, size, and anatomic location from the Impression section of the radiology report. They used CRF and MEMM (McCallum et al., 2000) models for extracting the tumor information such as tumor size, tumor count, anatomical part etc. A window size of 2 with unigram model is used for identifying the entities such as anatomy, tumor count, size etc. Once the entities are identified, it is further associated with relation to identifying if the disease is present or not. This model is a simpler variation of the model discussed in this thesis since it only identifies conditions related to one specific disease. Also, the entities identified in this approach are fairly straight forward which can be identified by using a dictionary. The overlap of entities are minimal which increase the accuracy of the system.

On similar note, another work done on radiology paper was to extract information on a structured format by Hassanpour and Langlotz (2016).They use NER from radiology reports based on 6 different types of classes. The process is based on the machine learning and NLP based methods. The model mainly deal with 3 different types of methods: Dictionary based named entity recognition, CMM based machine learning approach and CRF based machine learning approach. The main contribution of that research was it was able to extract clinically significant information from radiology report based on an information model which is radiology specific. The methods are based on existing machine learning approaches. The information can be further used for searching images of the report. The information model used for this paper was based on 5 different classes of clinical term data namely: anatomy, anatomy modifier, observation, observation modifier, and uncertainty. It is assumed that most of the clinically significant data of the radiological report are covered with the above information model.This information model was originally developed to support the system by Langlotz and Meininger (2000). The radiology dataset was adopted from RadCore[9] which has a collection of radiological reports from various sources. For the study, the reports were extracted from 3 different institutions: Mayo Clinic, MD

[8]https://en.wikipedia.org/wiki/Hepatocellular_carcinoma

[9]https://www.pennmedicine.org/departments-and-centers/department-of-radiology/radiology-research/core-facilities/radcore

Anderson Cancer Center, and Medical College of Wisconsin. And the chest CT (Cat Scan) reports are used for the training and testing of the model. The train and test reports were constructed based on manually annotated data from 150 chest CT reports. Chest CT reports have a variety of organs mentioned in the report and it is assumed that the reports are fairly complex.

The features extracted from the report consisted of the following segments:

- Part of speech tags: Extracted the POS tags for the phrases
- Word stems: Stemming the word to get its root form
- Word n-grams: Prefix and suffix substrings with less than 6 char in length is extracted
- Word shape: Orthographic signature of the word based on Stanford NLP toolkit
- Negation: Used Negex which is a negation extraction tool which identifies negations for phrases.
- RadLex[10] lexicon: Controlled lexicon for the radlex terminology.

The method used cTakes (Yetisgen-Yildiz et al., 2013) as the dictionary based method to identify the named entities of the reports. The dictionary is derived from the Radlex ontology. The terms were converted to their canonical forms before evaluating them. The Longest matching phrase is considered for the matching of dictionary entries. Further, they used CRF and CMM sequence classifier models to identify the entities.

Our model tried to design a system which can extract clinically significant information without focusing on any one specific disease or clinical data. The information extracted by our system can be used as a key information for bigger models which can be used for high-level patient profiling systems and in advanced machine learning tasks which use radiology data. We use robust features for training and modeling our system.

[10]https://www.rsna.org/RadLex.aspx

Chapter 3

Methodology

In this chapter, we discuss the methodology. The overall flow of the entire model is shown in Figure 3.1. In summary, the radiology reports are provided to the model as text files which are already de-identified of patient information. The reports are then cleaned of spelling and word-joined errors using customized algorithms. Once the reports are cleaned, they are provided to the feature extraction modules which extract the word and sentence level features which are used as input to the information extraction module. The information extraction module captures the features extracted from prior modules and builds a machine learning model which is used for predicting future report's information to be extracted. A document classification module is used for classifying the entire reports to positive or negative. All relevant output information is provided to user through a web interface which further allows the user to tweak the model.

In the following sections, a description of each step in the methodology is explained starting with document preprocessing module. We then explain in detail about the feature extraction module, information extraction module, document classification module and the final active adaptive interface provided to the user.

3.1 Document preparation

Real-world radiology reports are usually associated with spelling errors and joined words errors. In this section, we explain in detail about the type of errors found in radiology reports, the algorithm used for correcting the errors and the tweaks implemented to improve the accuracy of the system.

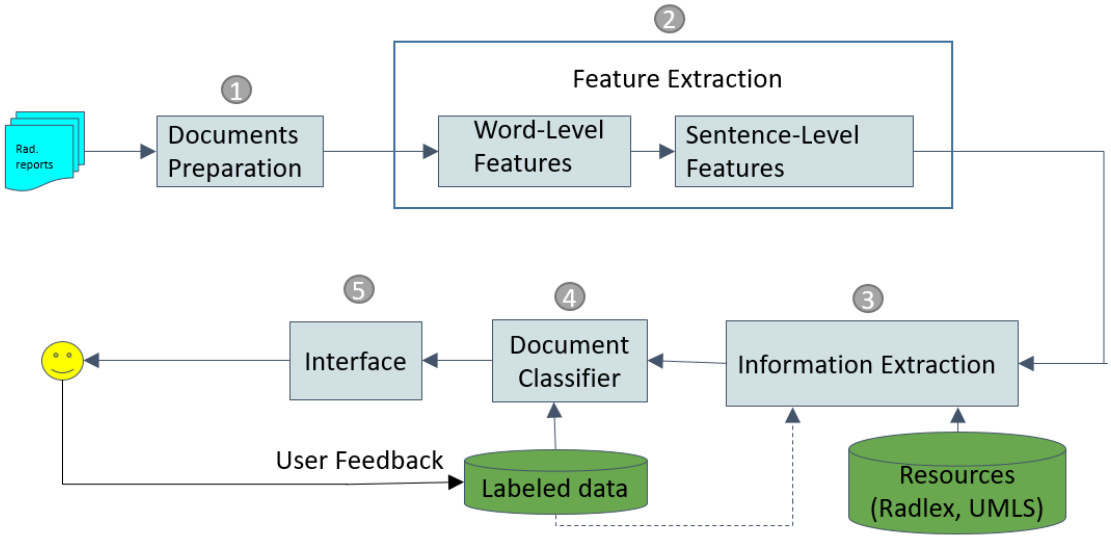

Figure 3.1: The overall view of the proposed system. At first all reports will be pre-processed (1) then several word and sentence level features will be extracted (2). The Information Extraction module (3) used the extracted features for identifying important phrases with their level of importance. The Document Classifier (4) classifies reports into two critical and non-critical categories based on the information exacted from the previous step. The visual interface (5) provides the user the extracted information and then tries to incorporate the user feedbacks in the system.

3.1.1 Common Errors

Radiology reports are generated by an automated system which converts the voice data from the radiologist to text files. It is required to review these automated records to check for spelling errors and should be corrected by a human reviewer. However, often times this process is not followed. One reason for this is that the doctor who uses these reports would be easily able to distinguish the errors and can easily review the radiology report. But a machine would not be able to easily identify the errors in the text. This is one of the major challenges in NLP. For processing the text document, we have to clean the text of the errors and unnecessary characters.

One of the most common errors in radiology text is the 'joined-word' error. Two or more words of the speech are represented as a single word in the text document. These type of errors are easily identified by the human because of the domain knowledge and language skills that we possess. An example of this error is 'thereare' which

should have been represented as ‘there are’. A sample radiology report with joined-word error is shown in Figure 3.2

PA and lateral chest x-ray. CLINICAL INFORMATION: 87-year-old male in emergency department withshortness of breath. Bilateral leg swelling. Productive cough. Historyof COPD. FINDINGS: Comparison is to multiple prior chest x-rays, the most recent beingfrom November 2010. The mediastinal contours are within normal limits. There iscardiomegaly, unchanged from prior examinations. There is nopneumothorax. There is vascular redistribution seen, in keeping withpulmonary edema. There is more focal airspace disease within the rightlower lung, new from the most recent examination in November but verysimilar in appearance to a more remote examination in November. Thereare small bilateral pleural effusions. The lungs are hyperinflated, inkeeping with the provided history of COPD. No fractures or aggressive bony lesions are seen. IMPRESSION: The vascular redistribution is in keeping with interstitial pulmonaryedema. The more focal airspace disease within the right lower lung islikely due to alveolar pulmonary edema given that a similar appearancehas been seen on prior examinations. A focus of infection cannot beentirely excluded.

Figure 3.2: A sample radiology report with Joined word error

Another type of error is the spelling errors. These are comparatively less to the joined-word errors. The main reason for this is that the voice o text system is less likely to make spell errors for common English words. And the reviewer would be more likely to correct spell errors when they see one.

3.1.2 Word Segmentation

The word segmentation model is based on the probability model to identify the possible corrected sequence. The algorithm is adopted from the book of Natural Language Corpus Data: Beautiful Data[1] (Segaran et al., 2009)[2]. The algorithm consists of a language model and an error model. The language model provides the probability of a word. The error model provides the probability of error given the word. For creating the word segmentation they have defined a model based on the Google n-gram values (Goldwater et al., 2009). And to reduce the computation and to improve the efficiency bi-gram model is used instead of using 3 or 5 grams.

[1]http://norvig.com/ngrams/
[2]http://norvig.com/ngrams/ch14.pdf

The language model: The language model is created as a probabilistic model based on the bi-gram data. The frequency table from the bi-gram data is used to calculate the probabilities. The parameters of the model are later learned from the corpus and for unknown words, a probability is estimated. The longer words are divided into smaller parts for applying the probability model. The probability of a given sequence of words can be obtained by the multiplying the probability each of the words in the sequence. However, in practice, multiplying all the prior word segments for a given input word can be costly because

- Language model will require a large amount of memory.(A 5-gram model require 30Gb of space)
- It is computationally expensive, a word of n characters would have 2n-1 candidates.
- The language model would have a lot of zero probability values since the candidate combinations are large and most of them are not real world word sequences. This requires back off algorithms.

The language model tries to split the given word into two parts and then calculate the overall probability as the product of the probability of the first word and the remaining segment. The best candidate would give the highest probability. The implementation of this method is done by using dynamic programing[3], which allows storing of the prior results for use on the next iteration. The complexity of this algorithm is $O(n^2)$. The unigram file contains only 1/3 of the million words which account for 98% of the most common tokens. The unknown word probability is created based on the length of the word so that an unknown word of length 100 would have less probability than an unknown word of length 5. The efficiency of this algorithm is improved by including the bi-gram values to the model, which would calculate the probability of a word based on the prior word.

$$P(W_{1:n}) = \prod_{k=1:n} P(W_k) \tag{3.1}$$

[3]https://en.wikipedia.org/wiki/Dynamic_programming

The unigram and bi-gram files used in this algorithm are from the Google web 1T corpora (Halevy et al., 2009) which can improve the performance of the system. The bi-grams are based on the Google bi-gram file and the model uses bi-grams which appear more than 100,000 times, which constitutes 250,000 bigrams. If a bi-gram is not present in the file, the model uses unigram model to calculate the probability. The algorithm uses Viterbi[4] method to efficiently compute the probabilities, where the complexity is $O(L^2n)$ where L is the length of the word and n is the number of segments.

Candidate enumeration: This step is used to enumerate all the possible candidates of the given word or a subsample based on careful analysis. For a word of length L, there would be 2*L-1 number of candidates.

Most probable candidate: This step will choose the most probable candidate based on the probability values. The candidates which provide the highest probability is chosen as the likely candidate.

$best = argmax_{c \in candidates} P(c)$

where c = Candidates , P(c) = Probability of the candidate

A sample radiology report after joined-word error correction is shown in Figure 3.3

3.1.3 Spell Error Correction

In common radiology reports, the chance of spelling errors is less compared with the joined word errors. This is because of the fact that, the reviewer is more likely to correct the spell errors in the review process. In this module, we try to correct spell errors which are missed by the reviewer.

The error model is implemented based on probability theory [5]. For each word of the sentence two probabilities are calculated. The probability of the corrected word

[4]https://en.wikipedia.org/wiki/Viterbi_algorithm

[5]http://norvig.com/spell-correct.html

PA and lateral chest xray . CLINICAL INFORMATION: is year old male in emergency department with shortness of breath . Bilateral leg swelling . Productive cough . history of COPD . FINDINGS: Comparison is to multiple prior chest xrays the most recent being from november 2010 . The mediastinal contours are within normal limits . there is cardiomegaly unchanged from prior examinations . there is no pneumothorax . there is vascular redistribution seen in keeping with pulmonary edema . there is more focal airspace disease within the right lower lung new from the most recent examination in november but very similar in appearance to a more remote examination in november . there are small bilateral pleural effusions . The lungs are hyperinflated in keeping with the provided history of COPD . No fractures or aggressive bony lesions are seen . IMPRESSION: The vascular redistribution is in keeping with interstitial pulmonary edema . The more focal airspace disease within the right lower lung is likely due to alveolar pulmonary edema given that a similar appearance has been seen on prior examinations . A focus of infection can not be entirely excluded .

Figure 3.3: Sample radiology report after Joined word error correction

P(c) and the probability of the corrected word given the current word $P(w|c)$. for each of the candidate, the candidate with the highest probability product is chosen as the corrected word, given by $corrected\ word = argmax_{c\ \in\ candidates}\, P(c)\ P(w|c)$. We used an existing implementation in python [6].

3.1.4 Contribution to Joined Word correction and Error Correction

It is not a good practice to use existing algorithms, which were designed for normal/ real-world text data for correcting a medical report. The occurrence of medical terms in the real world is much lower than the common words. So the system would produce inaccurate results for most of the medical terms. For example, 'nabothian' would be segmented into 'na' + 'both' + 'ian' since these separate words are more common than 'nabothian'. So we have modified the algorithm in two ways.

- Instead of using normal Google web 1T corpus (Lewis, 1998) with the first 333,000 uni-grams and 250,000 bi-grams (the black box algorithm uses this corpus to increase speed), we included medical or radiology term counts from the original Google n-gram corpus to the Google web 1T unigram corpus. This would make the algorithm assign a healthy count (occurrence in the real world) to the medical domain words rather than assigning a default unknown-word

[6] https://pypi.python.org/pypi/autocorrect/0.1.0

count. This would increase the probability value calculated for medical domain words.

- We used Radlex (Langlotz, 2006) and UMLS (Bodenreider, 2004) dictionaries which we created from the Radlex ontology and UMLS Ontology. Each word is checked in UMLS and Radlex dictionaries to see if it is a valid medical term. Only terms which are not present in these dictionaries are processed for joined word and spell error correction. This increased both the speed and accuracy of the word segmentation and spell correction system.

3.2 Feature Extraction

This section explains in detail regarding the auxiliary features used in training the machine learning models for extracting the clinically significant medical phrases from the reports. The first type of features are the word level features discussed in detail in Section 3.2.1 and the second type of features used are the sentence level features discussed in Section 3.2.2.

3.2.1 Word Level Feature Extraction

In this section, we discuss the various word level features extracted from the radiology reports which is further used for modeling the machine learning models. These features are used for identifying the various characteristics of a given word. More often than not, the words associated with medical domain have some characteristics compared to the normal words. The word level features try to capture the structural information of a word and use it for helping the machine learning model to determine the type of word processed.

Word level features are auxiliary features which are extracted from each of the words in the report. These features are used by the CRF model (Lafferty et al., 2001) in the final information extraction model. These features are explicitly created by us

for enhancing the performance of the model on identifying the criticality levels of the current word based on its structure. The various word level features extracted are:

- Stem and lemma of the word: The *stem* is the core part of a word. For example, the stem of playing is play. The *lemma* is the canonical or dictionary form of the word.

- Part of speech: We used the MedPost/SKR part-of-speech tagger (Smith et al., 2004) to extract the POS tags for our words.

- Word length: length of the word (number of characters).

- Anatomy: This is a boolean flag value which is set if the given word is an anatomical word. The anatomy dictionary for this flag is generated from the Radlex (Langlotz, 2006) ontology.

- Suffix and prefix: We extract the first and the last two letters of a word as a two-letter prefix and suffix. We also use the first and the last three letters of the word as three-letter prefixes and suffixes, respectively.

- Critical level flags: This is a boolean flag value which is set if the given word is a high-critical, critical or non-critical word. This dictionary is created based on the tagged data set generated by the human taggers.

- Meta Label and Meta Concept: This is the Meta Label and Meta Concept for a given word generated using the MetaMap system (Aronson, 2001).

- Filter words: The tagger automatically highlights several phrases to the human annotator, during the tagging process, based on the dictionary model. We capture explicitly the phrases which are removed by the human annotator during tagging process. These words are used to create a boolean flag feature which helps the system to eliminate some medical terms that are commonly disregarded by the emergency physicians.

3.2.2 Sentence Level Features

In this section of the thesis, we discuss the second type of features extracted from the radiology reports. Sentence level features capture the context of the given word. These features focus on the previous and next words of the current word in the sentence. These features are explicitly created by us for enhancing the performance of the model in identifying the criticality levels of the current word based on its context information.

The various sentence level features used are:

- Previous and next word Part of Speech tags: These features help to identify the type of the current word. Similarly to the word-level POS, the sentence-level POS tags are generated from the MedPost (Smith et al., 2004).

- Next Negative and next positive words: This feature identifies the positive or negative sentiment words after the current word. The positive and negative sentiment word list are extracted based on the social media sentiment analysis[7]. The value of this feature is the actual positive or negative sentiment of the word.

- Previous and next negative word positions relative to the current word: This feature calculates how far the negative word is located from the current word. The negation word-list in this feature is based on the Negex (Chapman et al., 2001) trigger word list. The value of this feature is the distance of the negative word from the current word.

- Word similarity: This feature compares the similarity of current word with the previous word. We used the word2vec (Mikolov et al., 2013) model for extracting this feature. The word2vec model was created based on 20,000 corrected radiology reports.

- Aggressive and Anatomy descriptors: These are boolean flags set to 1 if the anatomy or aggressive descriptors (from Radlex) are present in a 7-word window size of the current word (3 previous words + current word + 3 next words).

[7]https://github.com/jeffreybreen/twitter-sentiment-analysis-tutorial-201107

- High-flag, crit-flag, and non-crit flags: These flags check for the high-critical, critical, and non-critical word presence in the 7-word window size. These dictionaries are created based on the manual tagger.

3.3 Information Extraction

In this section of the thesis, we discuss the information extraction module of the system. This module is the heart of the system which extracts the clinically significant information and assigns critical values to them. We identify three different types of phrases from the radiology reports. The three types of phrases are high-critical, critical, and non-critical. We discuss two type of systems in this section. The first is a trivial dictionary based model which tries to identify the medical phrases. The second type model are the machine learning models which is implemented in the final interface.

3.3.1 Dictionary based model

The Dictionary based model is a trivial model which uses dictionaries extracted from the Radlex ontology. We extracted 4 dictionaries from Radlex. The dictionaries are based on the paper by Hassanpour (Hassanpour and Langlotz, 2016). The dictionaries are anatomy, modifier, observation, uncertainty. We further process the dictionary values to eliminate 'verb' words from the dictionary since they are usually not of importance to the doctors. Further, we also remove stop words from the dictionaries.

This model is used in the manual tagging interface provided for the user for tagging clinically significant phrases. It helps the user to easily focus on possible medical terminologies and conditions in the radiology reports. They help the user in long reports, where the chance of missing a medical condition is high. This model will not be able to identify the critical level for the medical condition because the medical condition criticality depends on its context information. In normal entity recognition systems, the chance of an entity belonging to multiple groups is less. But

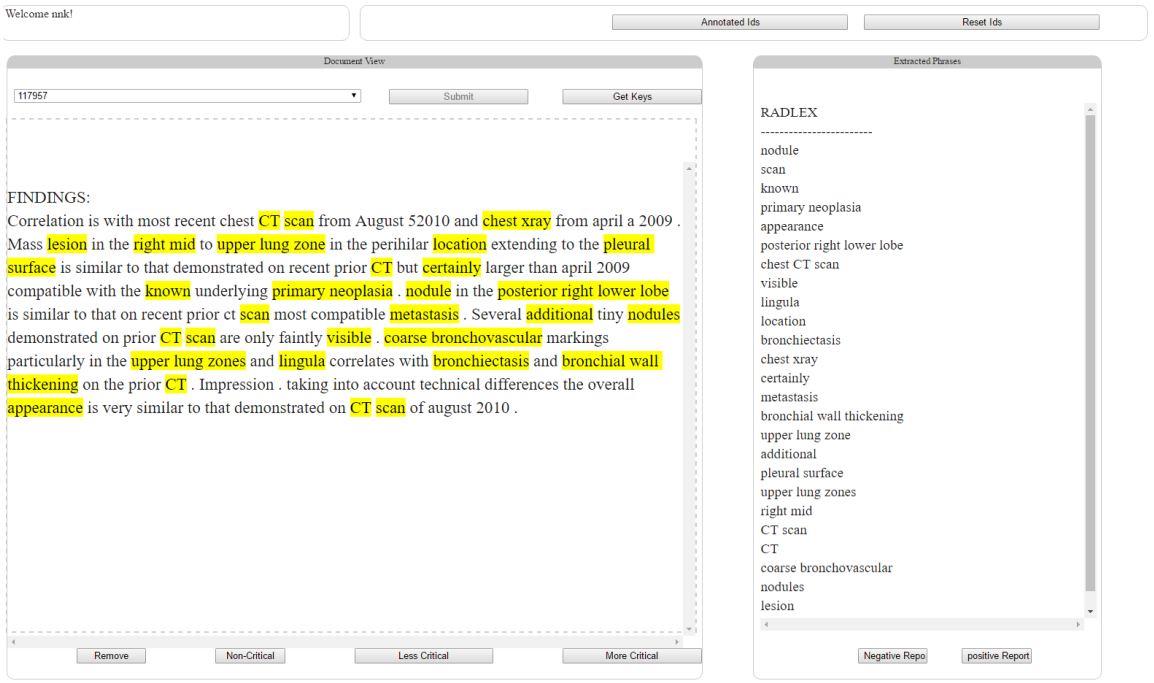

Figure 3.4: Dictionary based model Phrase extraction implemented on tagging interface

in our model, a disease can be in any of the 3 criticality levels based on the context. A sample of the Dictionary based phrase extraction is shown in Figure 3.4.

Tagging Interface

We used a simple web interface for the user to interact with the system. The interface is used for getting the manual tag data from the user during the initial training process. The interface for manual tagging consists of buttons which the user can use to mark the relevant information along with the different critical levels. It also provides the user with an option to mark the overall critical level of the report. By default, the phrases extracted using the dictionaries are assigned as non-critical and are highlighted. The user can change the criticality levels of both highlighted and non-highlighted phrases. The user also has the ability to correct any spell errors still present in the report.

3.3.2 CRF Model

Our objective in this thesis was to extract the relevant information about the patient's medical condition from the radiology reports and assign a critical value for them. Extracting the information from a sentence is similar to identifying the various entities in the sentence. In the area of natural language processing, it is usually achieved by sequence learning models. Given an input vector x that is divided into $x_0, x_1..x_T$ the sequence classifier produces a set of outputs y= $y_0, y_1, .y_T$ (Sutton and McCallum, 2010). Here each x_s contains various information about the word at positions. This information is discussed in Sections 3.2.1 and 3.2.2.

CRF combines the advantages of both classification and graphic model into a model which can leverage the multivariate data with the help of a large number of input features (Sutton and McCallum, 2010). This is precisely the reason that we chose CRF model for implementing our information extraction model. We provide several auxiliary features which are used by the model for predicting the class value for each word in the sentence. CRF also have other advantages over MEMM and stochastic grammars which have strong independence assumptions. CRF also performs better than MEMM and other discriminative graphic based models which have bias towards states with few successor states (Lafferty et al., 2001).

The formal definition of CRF is given as below (Sutton and McCallum, 2010) : Let Y, X be random vectors, $\theta = \{\theta_k\} \in R^K$ be a parameter vector, and $\{f_k(y,\ y0\ ,\ xt)\}_{k=1}^{K}$ be a set of real-valued feature functions. Then a linear-chain conditional random field is a distribution p(y—x) that takes the form

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} exp\ (\sum_{k=1}^{K} \theta_k f_k(y_t\ ,\ y_{t-1},\ x_t)\ ) \tag{3.2}$$

The graphical structure of various generative and discriminative models is shown in Figure 3.5. As we can see from the structure, MEMM and CRF are closely related. Each label Yt uses the word and sentence level features for its weight learning, and the process is shown in Figure 3.6. During training process, each label Yt uses the word

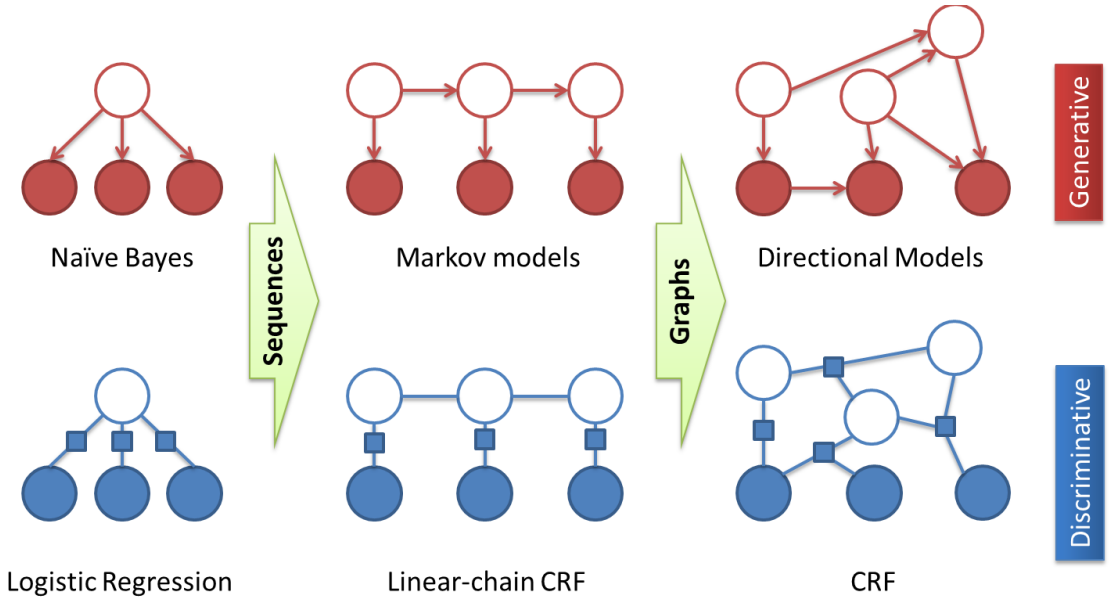

Figure 3.5: Diagram of the relationship between naive Bayes, logistic regression, MEMM, linear chain CRF, generative models, and general CRF (Sutton and McCallum, 2010)

level,sentence level, previous and next labels for learning the weight parameters. As discussed in Section 3.2.2, the sentence level features are extracted from the previous and next words of the sentence relative to current word. Here X values are the actual words in the sentence and Y values represent the labels (in this case criticality levels).

The model is trained to classify the phrases into three separate classes chosen after consulting with an emergency physician. Since the emergency physicians are primarily concerned with the immediate treatment of a patient's condition, it is necessary for the system to find medical phrases which have to be treated immediately. We use the classic BIO (Carreras et al., 2003) model for labeling the training data. Prefix B- indicates the beginning part of the phrase and I- indicates the subsequent words. eg: B-Crit and I-Crit labels are used to indicate critical phrases, and the phrase 'heart is enlarged' is labeled as 'B-Crit I-Crit I-Crit'.

Using the CRF model, we would be able to get the probability scores for the various labels for each word and the highest probability value is provided as output by

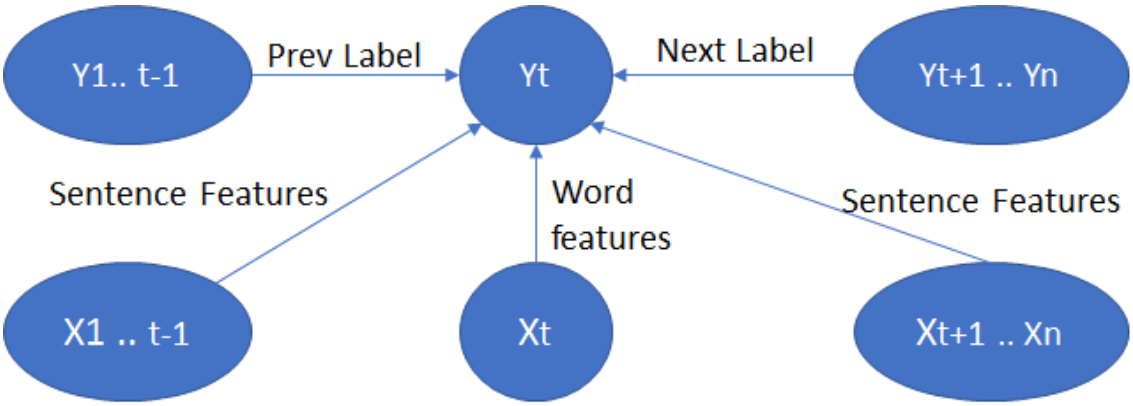

Figure 3.6: Feature function usage in the CRF model implemented for clinically significant information extraction. The sentence level feature functions looks into the previous and next words of current sentence while word level feature looks at the current word structure.

the CRF model. However, we can take advantage of the various probability values for evaluating the confidence score of the prediction. For instance, if the predicted label probability is 0.5, we can conclude that the model is not very confident on the predicted label, while a probability score of 0.9 shows high confidence. We used the L-BFGS (Nocedal, 1980) algorithm for the convergence during the training of CRF model. L-BFGS provides the fastest and accurate results compared with other algorithms.

3.3.3 Structured Perceptron

The second machine learning model implemented is the Structured Perceptron. We used the Structured Perceptron to compare our CRF model's performance with another generative model. We used the same auxiliary features that we used for CRF model in the Structured Perceptron as well. This allows us to directly compare the performance values for both the machine learning models.

Structured Perceptrons are generative models which are used for sequential classification with lots of features [8]. Similar to the CRF model, each feature in the Structured Perceptron is given a weight value. During the training process, the weight is increased for each of the positive samples and weight is reduced for negative samples, as $w \leftarrow w + y\,\varphi\,(x)$ where $\varphi\,(x)$ are the feature vector for the input X and Y is the output class and w is the weight assigned (Daumé III and Marcu, 2005). The highest incorrect prediction is given by Equation3.3:

$$\widehat{Y} = argmax_Y \sum_i w_i\,\phi_i\,(X\,, Y) \tag{3.3}$$

The weight values are then updated by equation3.4:

$$w \leftarrow w + \phi(X\,, Y') - \phi(X\,, \widehat{Y}) \tag{3.4}$$

If the highest scoring answer is the correct label, weight is not updated. Otherwise, the weight is reduced for that label. The algorithm is given in Algorithm 1[9]. More detailed explanation can be found in (Collins, 2002)

```
1 create map W;
2 for I iterations do
3     for each labeled pair X, Y_prime in the data do
4         Y_hat = HMM_VITERBI(W,X)
5         phi_prime = CREATE_FEATURES(X,Y_prime)
6         phi_hat = CREATE_FEATURES(X,Y_hat)
7         W += phi_prime - phi_hat
8     end
9 end
```

Algorithm 1: Structured Perceptron learning algorithm

[8] http://www.phontron.com/slides/nlp-programming-en-12-struct.pdf
[9] http://www.phontron.com/slides/nlp-programming-en-12-struct.pdf

3.4 Document Classifier

Once the radiology report is processed by the information extraction module, we classify the overall document to two classes. A positive report is a high priority report, while a negative report is a low priority report. This classification allows the system to highlight the high priority reports to the doctors and allows to prioritize the report. We used three well-known document classification algorithm to evaluate and compare the performance of the system. We classify the document using bag-of-words with tf-idf weights (Aizawa, 2003) and compare the results with the performance score of classifiers by providing the information extracted from the information extraction module.

We used RF, SVM with linear kernel and SGD for classifying the radiology reports. RFs are a special case of bootstrapping method in which n trees are generated from the sample and each un-pruned tree is based on m predictors. The final result is obtained by averaging the result of n trees (Liaw and Wiener, 2002). In SVMs the objective is to find a function that maximizes the margin between the two classes (Gunn et al., 1998). SGD works in a similar sense as the gradient descent method with lower steps taken each time. More detailed explanation can be found in (Amari, 1993). We used 3 algorithms to compare how the information extracted works for each of these completely different approaches to classification.

For each algorithm, we compared the precision, recall, and f1-score by using bag-of-words with tf-idf weights and then using criticality level phrases extracted using the CRF model. The models were trained using 10 fold cross validation and the average precision, recall and f1-score values of each model with the two types of features (bag-of-words with tf-idf weight and criticality level phrases extracted using CRF model) are compared. For training the algorithms with criticality level phrases extracted, we used a patient level report vector discussed in Section 3.4.1 along with some additional features such as word count of the given report, and criticality level phrase counts on each report.

3.4.1 Document Matrix

The Document Matrix is generated based on the output of the machine learning Model. It is essentially a vector for each of the patient reports where the columns represent the unique phrases extracted from all reports. This information can be used to quickly identify the condition of the patient from a collection of records. The level of criticality for each phrase is represented by a numeric value. A high-critical phrase is represented by +1, a critical level phrase is represented by 0.5 and a non-critical phrase is represented by -1. A sample snapshot of the Document Matrix is shown in Figure 3.7

	cardio mediastinal contour	pneumothorax	total shoulder prosthesis	linear densities	edema	Heart size	pulmonary vascular redistribution	pleural effusions	COPD
111887	0	0	0	0	0	0	0	0	0
120398	0	0	0	0	0	0	0	0	0
118793	0	0	0	0	0	0	-0.5	0	0
115771	0	0	0	0	0	0	0	0	0
113838	0	0	0	0	0	0	0	0	-0.5
114710	0	0	0	0	0	0	0	0	0
114098	-1	-1	-0.5	1	0	0	0	0	0
110570	0	0	0	0	-1	-1	-1	-1	-0.5

Figure 3.7: Sample Feature matrix for extracted medical phrases with critical levels. +1 is for high-critical phrase, 0.5 for critical and -1 for non-critical phrases. each row denotes each patient Ids. 0 values denotes 'not present in the report'.

3.5 Active Adaptive Interface

Our Active Adaptive Learning Interface is the user interface which shows the final phrases extracted and their criticality level to the user. This interface can be used to edit the extracted phrases predicted by the model. The user can add, remove, or change the criticality level of the phrases and the model is able to learn from the annotations of the report for predicting the phrases for next report. This is achieved by including the predicted phrases and criticality levels as part of the binary level auxiliary features. This helps the system to provide higher weight to the observed word based on corrected or previously predicted phrases. A sample screen shot is given in Figure 3.8.

The interface is also able to provide the level of certainty for the phrases as well as the overall criticality level of the report. The interface provides a visual cue for the

user for the terms which are less certain by the system (shown in larger font). The user has the ability to edit the tag (criticality level) of the phrases or leave it as it is. This extra information provides the user with the phrases which may have to be manually annotated by the user. We only focus on the critical (high-critical/critical) level phrases and the OTHER type of phrases of the radiology report for providing the uncertainty levels. We omitted uncertainty level for non-critical phrases since they are terms which are usually not of interest to by emergency physicians and to simplify the user's interactions. OTHER phrases are phrases which are not having any critical information (for example, medical phrases which are not tagged by the human annotator during training because it is not of much significance on the condition of the patient). OTHER phrases are focused since they are phrases which are perceived to be having no information by the system but can have valuable information to the user. The system determines a phrase as uncertain based on three cases as given in the list below.

- Its high-critical prediction probability is at least 0.1 and the predicted label is not high-critical.
- Critical prediction probability is at least 0.3 and the Predicted label is not critical.
- Predicted probability is less than 0.5 and the predicted label is Other.

The system also provides the user with the overall criticality level of the report as well as how confident the system is in its prediction. This can help doctors to identify emergency reports faster. A report is shown as low confidence prediction if the report class predicted distance is within one unit distance of the hyper plane. If the distance is more than one unit, it is predicted with high confidence. The distance score is negative for non-critical class and positive for the critical class.

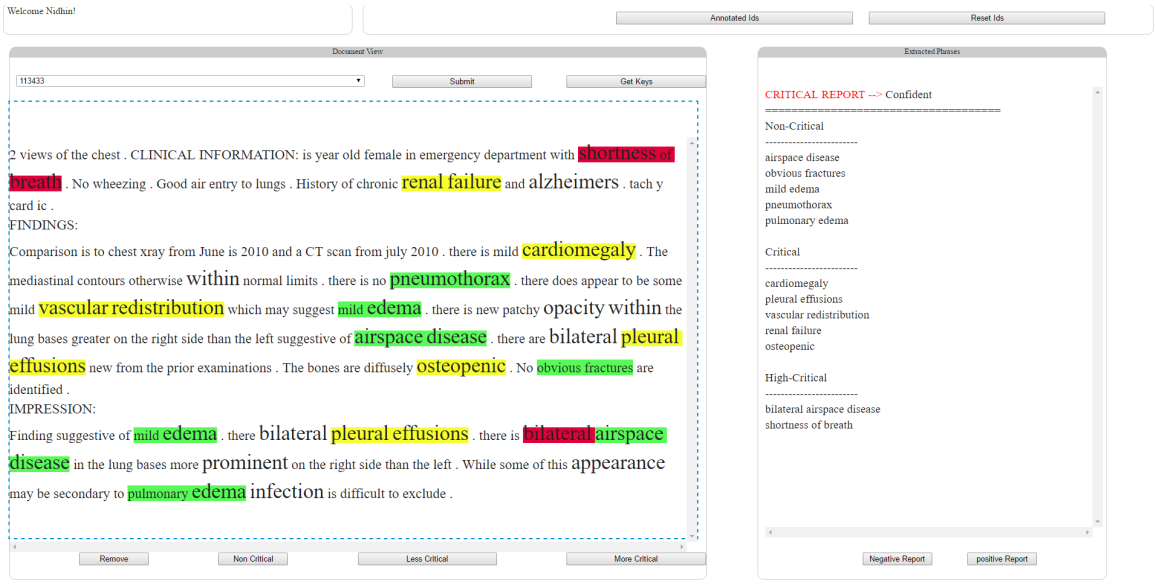

Figure 3.8: Final Interface which highlights the information extracted from the radiology reports along with critical levels for the phrases extracted. The overall document class (positive/negative) is shown at the upper top corner with the confidence level

Chapter 4

Experiments and Results

4.1 Data Collection

For the purpose of this thesis, the dataset used is a real world radiology data set. We focus on chest radiology reports because it is more complex and can help the system to be more robust. Chest radiology reports usually contain medical terminologies and conditions associated with chest, abdomen and sometimes part of the leg of the patient. This provides the model with a variety of medical terms and complexities associated with the report. The complete chest radiology dataset contains more than 20,000 radiology reports. These reports are created using voice to text processing systems and are reviewed later by a radiologist to correct any errors produced during the conversion. However, these radiology reports contain errors created by joined words. Our model corrects these errors during preprocessing stage so that the human annotator do not have to correct these errors during the tagging process.

In this thesis, our system is modeled to extract clinically significant information based on the critical level of medical conditions. We divided the critical levels to high-critical, critical and non-critical levels. Even though the initial model was based on chest radiology reports, it was later extended to model abdominal radiology reports.

The tagged data is collected using the manual tagging web interface which is coupled with the dictionary based phrase extraction model discussed in Section 3.3.1. The phrases are extracted based on the various dictionaries and are highlighted for tagging to the user. Each of the phrases is assumed to be non-critical by default. This allows the user to only focus on the critical and high-critical phrases and change the critical level using the appropriate buttons provided through the interface. The interface also captures the overall document class provided by the user. Once each

report is tagged, the user clicks on the overall class of the report (either a positive report or negative report) and the report with the tags are saved and removed from the interface. The user manual tagger interface also has the functionality to view previously tagged reports if needed. For this thesis, the user was able to tag 253 chest radiology reports. Special instructions were provided to the user to provide us with an equal distribution of both positive and negative reports and reports of varying length. To simplify this process, the interface was loaded with only 2000 reports chosen from the initial 20,000 with varying length of reports.

4.2 Evaluation Measures

We have used CRF and Structured Perceptron models for extracting information from the radiology reports and SGD, RF and SVM for classifying the radiology reports.

The precision is the number of true positives over the sum of the number of true positives and number of false positives as shown in Equation 4.1.

$$P = \frac{|T_p|}{|T_p| + |F_p|} \tag{4.1}$$

Recall is the number of true positives over sum of number of true positive and number of false negatives as shown in Equation 4.2

$$R = \frac{|T_p|}{|T_p| + |F_n|} \tag{4.2}$$

F-measure or F_1 is a single value representation for precision and recall, and it is a harmonic mean of precision and recall. The formula to calculate F_1 is show in Equation 4.3:

$$F_1 = 2\frac{P \times R}{P + R} \tag{4.3}$$

Accuracy is the number of correctly identified samples from the entire dataset and it is calculated as the sum of the number of true positives and true negatives over total number of samples $|D|$ as in Equation 4.4.

$$A = \frac{|T_p| + |T_n|}{|D|} \tag{4.4}$$

All the reported values are macro and weighted average. Macro-average is the average of values of the system on different sets and it is shown in Equation 4.5. Whereas in weighted average we consider imbalance in the number of samples of different sets and each set is assigned a weight as shown in Equation 4.6.

$$Macro\_average = \frac{\sum_{i=1}^{n} V_i}{|D_i, D_j, ..., D_n|} \quad where, \tag{4.5}$$

n is the total number of data sets, V_i *is the value of the data set* D_i*.*

$$Weighted\_average = \frac{\sum_{i=1}^{n} W_i.V_i}{|D_i, D_j, ..., D_n|} \quad where, \tag{4.6}$$

n is the total number of data sets, V_i *is the value and* W_i *is the weight of the data set* D_i*.*

4.3 Report Preprocessing

The most common error in the radiology reports were the joined words. The word segmentation module is used to segment the joined words present in the radiology reports. We used two methods to test our word segmentation module. Initially, we used a clean-text data set, which does not have any spelling errors, and we tested our model to check its accuracy. This provides us with an estimate of how many bogus word-segmentations are introduced, by the model, on clean text. For the second test, we created joined words (specifically radiology domain terms) and tested the system once again for the accuracy of segmentation.

For testing of the model with clean text, we used the text8 dataset (Zhang et al., 2016) which contains over 3 million words. The text8 data is given to the algorithm for processing and we checked the number of words which are segmented by the model (ideally it should be 0). We obtained an accuracy of 98.9% on this data. This test was done to make sure that the algorithm does not segment correct words present in real world documents.

For the second test, we created joined words from the words present in the Radlex ontology, chosen randomly and then combined together to create a joined word. The words chosen are medical words (not common words) in order to provide a better

Table 4.1: Accuracy of base and implemented models for joined word correction

	Accuracy (Our model)	**Accuracy (Existing model)**
Text 8 Dataset	98.9%	98.9%
Radlex random word combination 10k iterations	87.46% (2w) 81.28% (3W)	42.58% (2w) 25.69% (3W)

view of how well the system performs on uncommon words. We tested 2-word and 3-word combinations. The experiment was repeated for 10,000 iterations. We obtained an accuracy of 87.46% for 2-word combinations and 81.28% for 3-word combinations. This higher accuracy was obtained after adding the unigrams from the Google n-gram corpus for radiology terms (explained in detail on Section 3.1.2). Without adding the unigram radiology terms to the algorithm, the accuracy of 2-word combination was 42.58% and for 3-word combinations, it was 25.69%. This clearly shows that our model, with the addition of radiology terms, provides the best accuracy results. The results are shown in Table 4.1

For evaluating the performance of the spell correction algorithm, it was tested with the text8 data set which has 3M words. It was found that the algorithm produces an error rate of 0.5%.

4.4 Inter annotation Score

In order to compare our model to real-world human annotation performance we asked a second annotator to annotate the radiology reports and we then examined the consistency between the two sets of annotations.

The second annotator annotated 57 random reports out of the 253 reports tagged by the first annotator. For calculating the inter-annotator score, we used two methods. First, we used a 'soft' matching algorithm that only calculates the inter-annotator agreement on phrases which were annotated by both annotators. For the second method, we calculated the Precision, Recall, and f1-score of the second annotator on annotating the reports by keeping Annotator-1 as the gold standard. In both of these

Table 4.2: Confusion matrix for annotations done by annotator two on the radiology reports. Gold standard is based on the initial tagging done by annotator one.

		Predicted					
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**B-HighCrit**	**I-HighCrit**
Actual Labels	**B-NonCrit**	129	7	6	0	13	0
	I-NonCrit	7	126	0	5	1	10
	B-Crit	11	1	16	8	33	5
	I-Crit	1	12	0	15	3	17
	B-HighCrit	8	0	8	0	95	9
	I-HighCrit	0	2	0	8	7	75

methods, we used the 57 reports annotated by the second annotator (Annotator-2).

The first evaluation method involves the calculation of the soft agreement score between annotators. The formula for the soft agreement score calculation is given in Equation 4.7.

$$Soft\ score\ =\ AVG\left(\sum_{i=1}^{57}\frac{W_i}{N_i}\right) \tag{4.7}$$

- W_i = Number of words predicted by both annotators with same criticality level in report i.
- N_i = Number of words predicted by both annotators in report i.

We obtained the soft agreement score of 71.47% on annotation. This proves that annotating a report and providing criticality levels to the phrases is a complicated task even for a human annotator who has ample domain knowledge. Moreover, reducing the annotation task to a 2-class system (critical/ non-critical) increased the inter annotation score to 85.01%. This experiment proves that the boundary of critical and high-critical can change based on the user's perception of each report. The confusion matrix for the soft score is shown in Table 4.2

The second evaluation method involves the training of the CRF model on the 200 reports that were not tagged by the second annotator. Once we trained the CRF model, we tested the model on the 57 reports tagged by the first annotator. We compared this result with the performance score obtained by asking the second annotator

Table 4.3: Precision, Recall and f1-Score comparison between human annotator and CRF model

	HUMAN ANNOTATOR			CRF MODEL		
	precision	**recall**	**f1-score**	**precision**	**recall**	**f1-score**
B-NonCrit	0.6825	0.6324	0.6565	0.8140	0.5122	0.6287
I-NonCrit	0.7241	0.6632	0.6923	0.8947	0.6041	0.7212
B-Crit	0.2712	0.1928	0.2254	0.4583	0.4074	0.4314
I-Crit	0.2239	0.2500	0.2362	0.4643	0.2203	0.2989
B-HighCrit	0.5220	0.7308	0.6090	0.6406	0.3228	0.4293
I-HighCrit	0.5682	0.7353	0.6410	0.7692	0.4000	0.5263
Average	0.5703	0.5930	0.5759	0.7359	0.4564	0.5601

to tag the same 57 reports. The results are shown in Table 4.3. The CRF model gives similar performance to that of the human annotator but with higher precision. The performance dip in f1-score is due to the lower recall value, which would improve on an ongoing basis as the system acquires more data.

Both these experiments proved that the annotation of clinically significant information and assigning critical values to them is a complex task. Even for a human expert who has years of experience, the significance of a medical phrase can change based on their own viewpoint. Another important point to be noted here is that the reports used for training the model are de-identified which limits the performance of the model. This is because of some information such as the age of the patient, sex of the patient and physical condition of the patient.

4.5 Machine learning models

We used two sequence classifiers for our phrase extraction and criticality level identification. For each of the criticality levels, we used separate labels. For non-critical terms, we used B-NonCrit and I-NonCrit as the labels (Beginning word and subsequent word). Similarly, we used B-HighCrit, I-HighCrit, B-Crit, I-Crit respectively for high-critical and critical phrases. We used Conditional Random Field and Structured Perceptron as our two machine learning sequence classifiers.

4.5.1 CRF model

The train data set for the machine learning CRF model is obtained from annotator one through the user interface implemented with the dictionary based phrase extraction model. The interface highlights medical phrases identified through dictionary lookup. The user (in this case, emergency physician) assigns a critical value to the phrases (can be phrases highlighted by the interface or phrases which the physician thinks are important for diagnosis). Each of the phrases annotated is given a critical score and saved as a data set. High-critical phrases have a score of +1, critical phrases have a score of +0.5 and non-critical phrases have a score of -1. These score values are later used for predicting overall Class of the report.

In this study, the doctor was able to provide 253 radiology reports, with varying levels of length and complexity in the reports. We used the already existing fast implementation of CRF (Lafferty et al., 2001; Sha and Pereira, 2003) for our Model. The features used for the CRF are discussed in Section 3.2.1 and 3.2.2. We used L-BFGS (Nocedal, 1980) algorithm for the optimization. The coefficient values are dynamically calculated based on the training data.

We used 10 fold cross validation (Refaeilzadeh et al., 2009) on the training data. Since we do not check for inter-sentence parameters, the algorithm treats each sentence as single sample. We have obtained an average f1-score (Sokolova et al., 2006) of 0.75.

Evaluating the performance of phrases extracted

Comparing the performance of the model on extracting various critical levels from the radiology report, we can see that the system performs well for all 3 types of the phrases. The system performs especially well on extracting high-critical phrases which provide extremely critical information to the doctors. Out of the 3 different types of phrases, the system performs comparatively less for the critical phrases. The main reason for this gap is the difficulty on segregating between critical and high-critical phrases. It also depends on the quality of the dataset and annotations. Since the

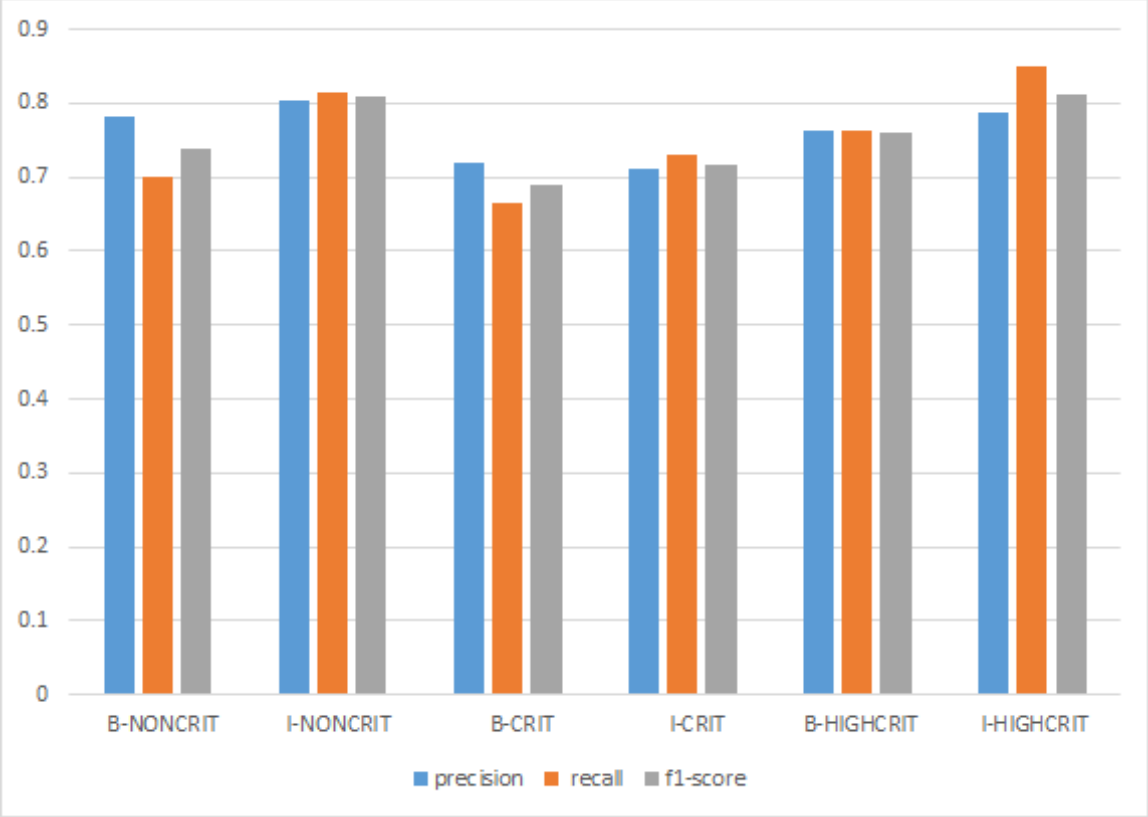

Figure 4.1: Performance of CRF model on extracting various critical information from radiology reports. B-NonCrit, I-NonCrit represents non-critical phrases beginning and middle words, B-Crit, I-Crit represents critical phrases and B-HighCrit, I-HighCrit represents high-critical phrases extracted from the radiology reports.Performance is measured by precision, recall and f1-score matrices.

annotation was done over a period of time, the doctors may tend to change their way of annotation on later reports. This may affect the consistency of the tagging. For example, the doctor may tag heart is enlarged as high-critical in earlier reports and may tag the same as critical in later reports. Figure 4.1 shows the precision, recall and f1-scores measured for CRF model.

The confusion matrix obtained on training the CRF model is shown in Table 4.4. The confusion matrix is based on the 10 fold cross-validated result obtained on training the model. Analyzing the confusion matrix, we can clearly see how the classifier performs for each of the critical level phrases extracted. The majority of the error

Table 4.4: Confusion matrix of CRF model with various critical level phrases. 'O' denotes phrases which are not irrelevant or are considered of no value to the doctors.'B' and 'I' denotes the beginning and Intermediate words of the phrase.

		Predicted labels						
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**B-HighCrit**	**I-HighCrit**	**O**
Actual Labels	**B-NonCrit**	63	4	2	1	2	0	21
	I-NonCrit	1	61	1	2	0	1	9
	B-Crit	2	0	35	1	5	1	8
	I-Crit	0	1	1	27	0	3	4
	B-HighCrit	2	0	4	0	40	2	2
	I-HighCrit	0	1	0	1	1	32	2
	O	12	11	5	5	3	2	1354

occurred is the misclassification of labels to one of the other critical levels. Misclassification to 'other' category is comparatively less.

4.5.2 Structured Perceptron

The second sequence learning classifier used in this thesis is the Structured Perceptron. Structured Perceptrons works similarly to other sequence classifiers such as MEMM (McCallum et al., 2000) and MEMMs (Rabiner and Juang, 1986) (Baldridge et al., 2010) and CRF. They also can be trained based on auxiliary features which allow the classifier to predict values based on the context information. The auxiliary features are used to learn the various weight values which are further used to predict phrases in unknown reports.We evaluated the Structured Perceptron using 10 fold cross validation and have obtained an average f1-score of 0.72 which is slightly less than the performance obtained from the CRF model. We used Structured Perceptron model to test the efficiency of our auxiliary features as well as to compare and evaluate the best model for the phrase extraction.

Performance evaluation of Structured Perceptron

The performance scores for Structured Perceptron model on extracting various critical level phrases from the radiology reports is shown in the Figure 4.2. Similar to CRF model, the performance of the Structured Perceptron is less on extracting critical

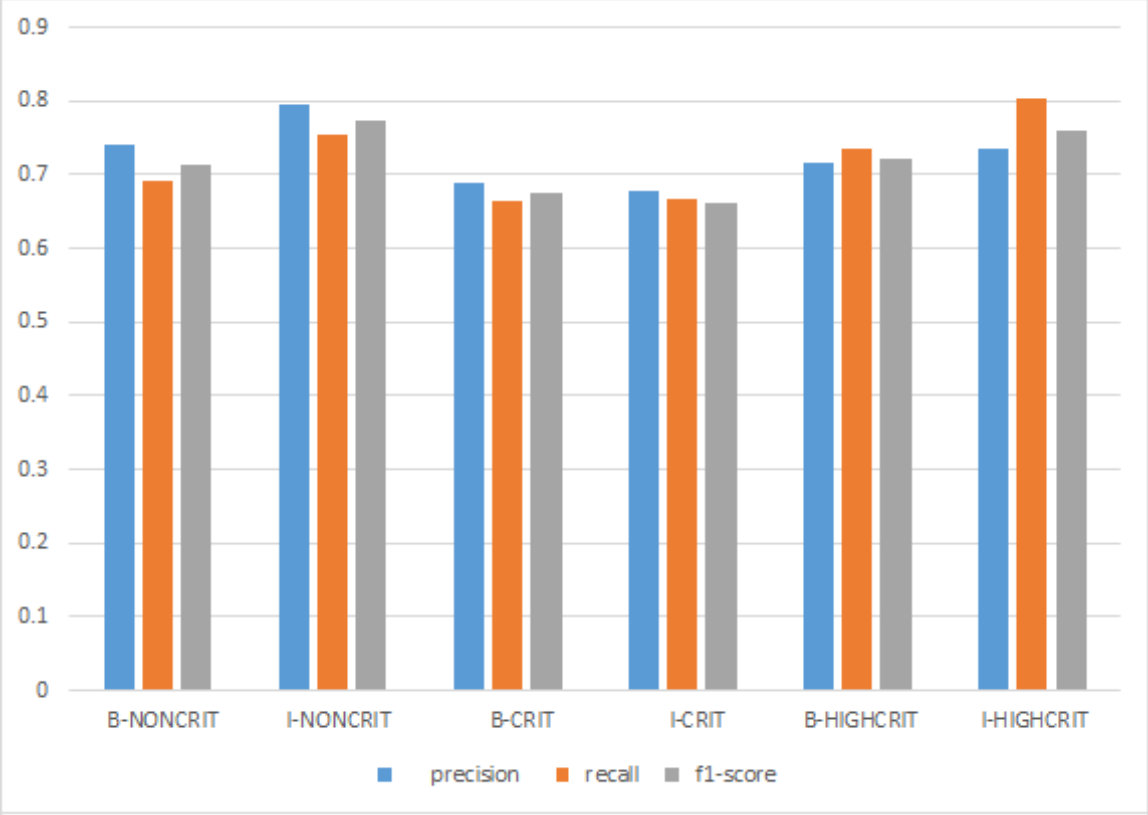

Figure 4.2: Performance of Structured Perceptron model on extracting various critical information from radiology reports. B-NonCrit, I-NonCrit represents non-critical phrases beginning and middle words, B-Crit, I-Crit represents critical phrases and B-HighCrit, I-HighCrit represents high-critical phrases extracted from the radiology reports.Performance is measured by precision, recall and f1-score matrices.

phrases compared to high-critical and non-critical phrases. The performance for non-critical terms were worse than CRF model (0.68). The f1-score for non-critical terms were almost the same as CRF model (0.76). And for high-critical terms, the accuracy was lower compared to CRF model (0.76).

4.5.3 Comparing CRF and Structured Perceptron Model

Both of these models are similar in performance. However, the CRF model performs better on average. The auxiliary features used for the training and prediction of sequence labels are the same. The CRF model is able to provide better recall than Structured Perceptron. Moreover, CRF provides the predicted probability values for the labels which can then be used for identifying the uncertainty of the predicted

values. Evaluating the results for both CRF and Structured Perceptron by t-test, we obtain a p-value of 0.0549.

4.6 Auxiliary Features and weights

For our machine learning models, we have used two types of auxiliary feature:, word level, and sentence level features. In this section, we compare the models' performance based on the auxiliary features provided. For the sentence level features, we have segmented the performance graph into two parts, namely sentence level features, and flag level (or binary) features. The binary features are provided separately since the contribution of the binary features on the models' performance is significant.

As we compare the performance of the system based on the set of auxiliary features, the sentence and binary-level features provide a more significant contribution to the models' performance than do word-level features. One reason for this difference is that some of the word level features are inherently present in the sentence level features as well. For example, previous and next POS tags give similar contributions to assigning the POS tags of the current word. We assigned the current word POS tag contributes to the models' performance in special cases such as the beginning and end words of the sentence, and one-word sentences where there are no previous or next POS tags.

Binary features are part of the sentence-level feature-extraction module. These features are the main contributors to the Machine Learning model used in the active adaptive interface. These features are dynamically created based on the prior-tagged reports. For example, tagged phrases provided by humans during the training process are updated dynamically as the user uses the active/adaptive learning interface. These features create dictionaries based on the types (high-critical, critical and non-critical) of tagged critical phrases. These features help the model to identify medical terms which are critical or high-critical on most of the reports.

The combination of the 3 sets of features provides the best accuracy results for

Table 4.5: Performance scores for CRF model based on various features used during training process.

		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit
Word Level	**precision**	0.656	0.676	0.647	0.577	0.569	0.625
	recall	0.625	0.797	0.584	0.546	0.501	0.613
	f1-score	0.639	0.731	0.610	0.541	0.529	0.611
Sentence Level	**precision**	0.756	0.768	0.601	0.531	0.653	0.681
	recall	0.683	0.739	0.483	0.395	0.511	0.579
	f1-score	0.717	0.752	0.532	0.439	0.571	0.620
Binary Level	**precision**	0.706	0.735	0.738	0.704	0.712	0.755
	recall	0.593	0.769	0.591	0.698	0.679	0.790
	f1-score	0.643	0.750	0.653	0.694	0.692	0.769
Combined	**precision**	0.781	0.804	0.720	0.712	0.762	0.788
	recall	0.702	0.816	0.666	0.731	0.762	0.850
	f1-score	0.737	0.808	0.689	0.716	0.760	0.811

our model. The sentence level features help in increasing the recall value of our model while the word level features are used to increase the precision of our model. Another reason for adding binary features and sentence level features is to increase the efficiency of the model on predicting medical phrases which are not seen by the model in previous training samples. The binary features ensure that the phrases previously were seen are given higher weight while the sentence level features ensure that the medical phrases which are not seen in the past are considered by the model. The f1-score comparison for various features is provided in Table 4.5.

4.6.1 Top Auxiliary Features

As mentioned in Section 3.3.2, the CRF model learns weights for the different auxiliary features provided to it during the training process. The new phrases are predicted based on the calculated feature values and weights computed during the prediction process. The top 5 feature weights used by the CRF model for predicting the new critical phrases is shown in the Table 4.6 .

On closer inspection, we can see that the binary features are a critical part of the model. They are almost always part of the top 5 positive features when predicting a critical phrase. Some of the other interesting features are the suffix words, which is part of the top features in predicting high-critical Values. Word-to-vec model plays another crucial role in predicting phrases. It is mostly used in the intermediate word

Table 4.6: Top 5 positive and negative features used for predicting the critical level phrases in the CRF model. 'B' denotes the beginning and 'I' denotes the intermediate words for a given phrase.

	Positive Features	**Negative Features**
B-High	'highFlag:True' 'nextWord1:can' 'prevWord1:lobe' 'suffix1:is' 'prevWord1:a'	'nextWordPos1:adj' 'prevWordPos1:det' 'pref2:bre' 'critFlag:True' 'nonCritFlag:True'
I-High	'word2VecSimilarityPrev:0.999597393253' 'nextPos:superior' 'highFlag:True' 'word2VecSimilarityPrev:0.999418976773' 'suffix1:in'	'pos_tag:adj' 'nextWord1:at' 'pos_tag:verb' 'pref1:le' metaConcept:'Qualitative,Concept'
B-Crit	'nextNeg:chronic,' 'prevWord1:mild' metaConcept:'Pathologic,Function"Finding' 'nextWord1:areas' 'prevWord1:known'	'pref1:ch' 'word2VecSimilarityPrev:0.999468437275', pref2:con' 'nextWord1:and' 'nonCritFlag:True'
I-Crit	'critFlag:True' 'pos_tag:noun' 'nextWord1:unchanged' 'word2VecSimilarityPrev:0.999558585312' 'word2VecSimilarityPrev:0.999675685209'	'pos_tag:adj' 'negExNext:' 'pref1:lu' 'suffix1:al' 'prevWord1:unfolding'
B-NonCrit	'word2VecSimilarityPrev:0.999108931789' 'negExPrev:1' 'prevWord1:consolidation' 'nextPos:appreciated,' 'prevWord1:based'	'prevWord1:a' 'suffix1:ly' 'negExPrev:15' 'pref2:int' 'suffix1:er'
I-NonCrit	'nextPos:improvement,' 'nextWord1:pattern' 'nextWord1:base' 'nonCritFlag:True' 'nextNeg:no no,'	'word2VecSimilarityPrev:0.998901530601' 'nextWord1:mediastinum' 'pref1:ma' 'suffix1:um' 'nextWord1:effusion'

predictions because it provides a close relation with the previous word. As expected, the next positive or next negative word feature provides information to the model on what the criticality level of the phrase is. For example, the word 'improvement' after the current word suggests that the word is possibly a non-critical phase.

Negative features help the model to decide if the current word is not a part of the critical phrase. Part of speech tags of the current word plays a crucial role in the negative features. Another important negative feature for predicting critical level are the previous, next words and prefix/suffix terms. The Binary features do not play a significant role in negative features.

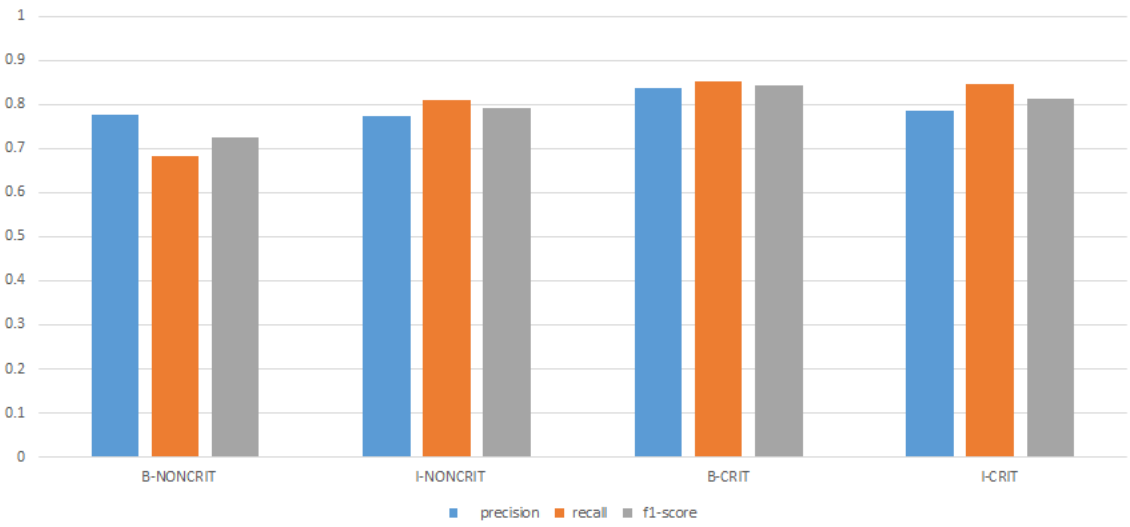

Figure 4.3: Performance of the CRF model when trained based on two critical level classes. High-critical and critical level phrases are joined together to produce a single critical Class.

4.7 Two Class Model for phrase Extraction

From our previous experiments, we found that the most prominent errors occurred for the machine learning as well as the human taggers is the misclassification of the phrases to critical and high-critical segments. A lot of factors contribute to assigning a phrase to high-critical or critical. So we combined the high-critical and critical segments into a single phrase Class and trained our model based on the two final classes

Table 4.7: Confusion Matrix for CRF model with two levels of critical phrases. High-critical and critical level phrases are merged into a single critical Level.'O' Denotes 'Other' or not tagged phrases.

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	64	4	5	1	20
	I-NonCrit	1	61	1	3	9
	B-Crit	4	0	87	4	8
	I-Crit	0	2	2	61	8
	O	14	12	9	9	1350

as non-critical and critical.

As expected the performance of the model increased to an average f1 score of 79.5 for the CRF model. The label performance of the CRF model is shown in the Figure 4.3. The performance of the system is especially High for the critical level phrases with an average f1 score of 82.7. However, this model is oriented more towards the general information extraction of medical phrases from the radiology report rather than the special information extraction required by the emergency physicians. The confusion Matrix of the two class model is shown in Table 4.7.

4.8 Scalability of the model

Once the CRF model is trained on the 250 reports, we obtain an initial information extraction model. We tested the model on unknown reports where we do not have labeled information and checked to see how the number of unique phrases extracted changes as we provide more reports to the model.

The number of unique phrases extracted increases linearly if we provide the feedback option turned on, as shown in the Figure 4.4. By using the feedback option, the system recognizes the phrases it has already predicted from each of the previous reports on the go and adds the phrases as part of the binary features for critical, high-critical and non-critical medical terms and uses this data for predicting critical phrases on the new report. This method is helpful if we have a good number of

labeled data for the system to train in as the error rate would be small.

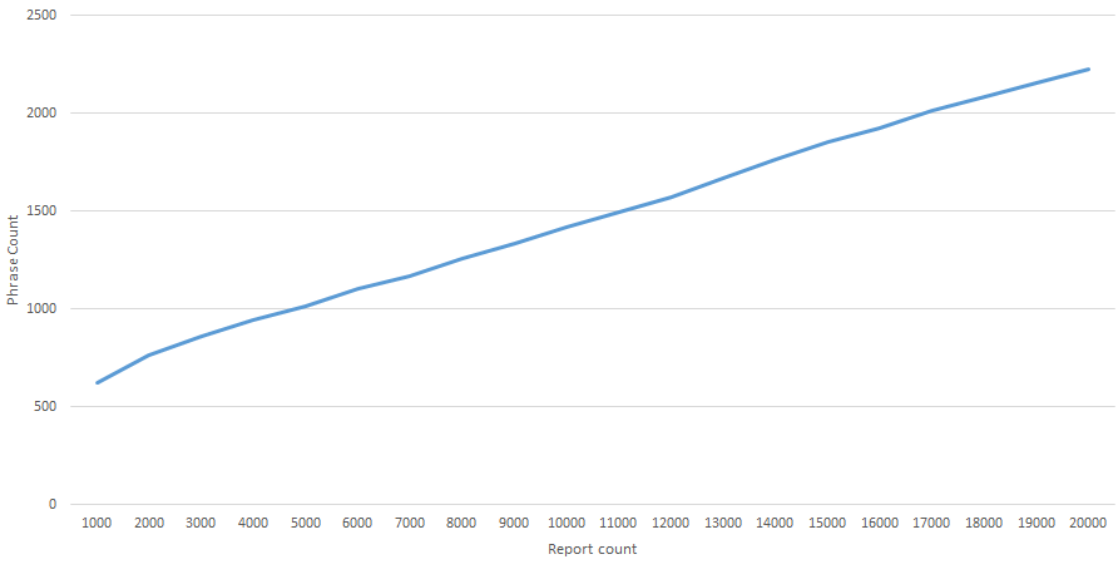

Figure 4.4: Number of phrases extracted by the CRF model when evaluated against unknown reports with feedback option. The phrases extracted from the first report is added to the binary feature dictionary and is used for the auxiliary feature creation of next report and so on.

4.9 Report Classification

The radiology reports are classified into two classes, critical reports, and non-critical reports. The classification is based on the overall report and is related to whether immediate action is required, on the patient in the emergency department. In order to analyze the relevance of the extracted phrases using the CRF model, we compared the classification accuracy of well known machine-learning algorithms using two methods. On the first trial, the reports are classified based on the 'bag of words' method having tf-idf weights assigned on those given in the report. In the second method, we used the phrases extracted using the CRF model along with the values assigned (-1 for non-critical phrases, 0.5 for critical phrases and 1 for high-critical phrases). We have used three separate machine learning algorithms (Linear Support Vector Machine, RF and Stochastic Gradient Descent from the Sklearn library[1]) to

[1]www.scikit-learn.org

compare the performance of each machine learning algorithm on these two types of feature. The comparison results are shown in Figure 4.5.

Comparing the results of the three algorithms on the two types of feature, we can see that the phrases extracted out-perform the 'bag of words' method having the tf-idf weights-based model on both the RF (Liaw and Wiener, 2002) and Linear SVM (Gunn et al., 1998). Even on the SGD (Amari, 1993) the phrases extracted have similar performance to 'bag of words' having a tf-idf weights model. Also, the phrases extracted from the reports are comparatively much fewer than 'bag of words' having a tf-idf weights model.

Using the phrases extracted we were able to achieve an average f1-score of 86.42, in comparison to the average f1-score of 86.52 for 'bag of words' having a tf-idf weights model with SGD. These results demonstrate that the phrases extracted from the radiology reports are quite powerful features in classification.

Evaluating the statistical significance using the student t-test on the results, we have obtained a p-value of 9.43E-08 and 1.1E-05 respectively, for random forest and linear SVM and for 'bag of words' having tf-idf weights model and extracted phrase features. These results show that the 'extracted phrases' method performs better on classification of the report using these algorithms.

4.10 Error Analysis

We have analyzed the misclassification errors for the CRF model which used the three level criticality levels for the extracted phrases. Upon analysis, the greatest misclassification occurs on classifying the non-critical phrases, which get classified as Other. These types of error are not a big concern in emergency-room practice since the doctors are mostly concerned about critical phrases. Even on manual tagging, depending on the report, some of the medical phrases may not be tagged by the doctor as non-critical. On analyzing the results, about 22% of the total non-critical phrases were predicted as 'Other' by the system. However, less than 2% of those terms which

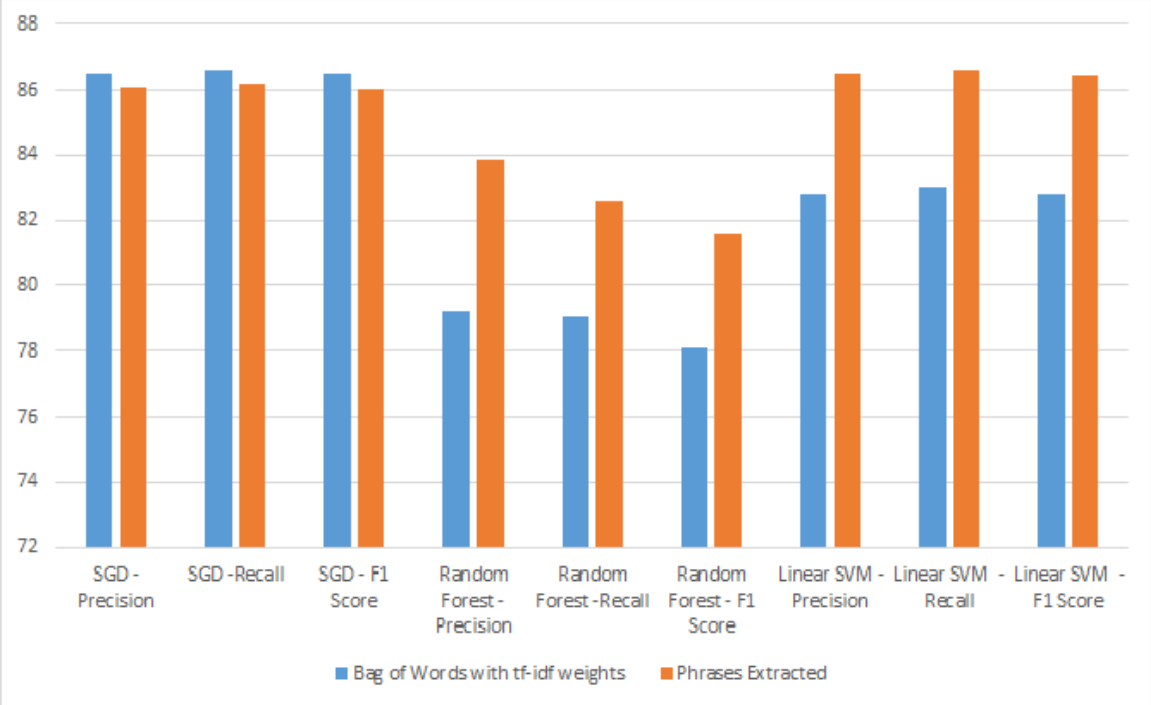

Figure 4.5: Performance comparison for various machine learning document classifiers on classifying radiology reports. Reports are classified to critical or non-critical classes. Each classifier is evaluated based on two type of feature sets. One is the tf-idf score of the words of the report and the second is using the phrases extracted using CRF model.

were actually non-critical were predicted as critical by the system.

Analyzing the critical phrases, the most common misclassification was, again, the classification of a 'critical' phrase as being 'Other'. However, the misclassification rate is lower compared to the non-critical phrases. The misclassification of critical phrases as Other is about 15%. However, on further analysis, it has been identified that the same phrase is misclassified in multiple reports which adds to the misclassification percentage. For example, the phrase 'Intrathoracic' is misclassified more than once, which adds to the misclassification rate even though only one phrase is misclassified. But this problem can be solved as we increase the amount of training data. As the doctors use the active adaptive learning interface through the on-line interface, these types of errors could be reduced considerably.

Table 4.8: Error rates for various critical level phrases extracted using the CRF model. The values are shown as percentages.

		Predicted						
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**B-HighCrit**	**I-HighCrit**	**O**
Actual	**B-NonCrit**	67.5%	4.7%	2.1%	0.5%	2.3%	0.0%	22.8%
	I-NonCrit	1.9%	80.6%	1.2%	2.0%	0.5%	1.7%	12.1%
	B-Crit	3.7%	0.0%	67.8%	1.9%	9.7%	1.8%	15.0%
	I-Crit	0.0%	3.6%	2.2%	76.2%	0.0%	7.0%	10.9%
	B-HighCrit	3.9%	0.2%	8.1%	0.4%	79.7%	3.4%	4.3%
	I-HighCrit	0.0%	1.9%	0.8%	2.4%	3.8%	85.3%	5.9%
	O	0.9%	0.8%	0.4%	0.4%	0.2%	0.2%	97.2%

Finally, for high-critical terms, the most common errors are misclassification of the criticality level. About 8% of the high-critical phrases are misclassified as critical phrases by the system. However, since the doctors are able to view both critical and high-critical phrases in the interface, along with the reports, these errors would not have a significant impact on the user experience. The various error types and the impact of them on the model is shown in the Table 4.8.

Chapter 5

Conclusion & Discussion

Radiology reports are complex reports which are often difficult to process using standard NLP tools because of the spelling and word-join errors. We propose a system that provides better accuracy in correcting joined word and spell errors, followed by extraction of medical phrases and their criticality level, and classification of the whole report as critical or not.

Information extraction from the radiology reports as phrases is complex but valuable data which can be further used in complex or simple applications. The complexity of this task is mainly centered around the criticality level which have to be assigned based on the context of the phrases extracted. The radiology report model extracts medical phrases and the associated criticality level (high-critical, critical and non-critical). This is accomplished by a CRF model that is trained on a small corpus of reports labeled by two emergency physicians. We demonstrate that this information extraction task achieves performance that is comparable to the inter-annotator agreement. For obtaining this performance we have developed advanced auxiliary feature extraction which extracts several types of feature values for each word to be predicted. The auxiliary features are broadly segmented into Word level and sentence level features in which the word level features provide information regarding the structure of the word including the part of speech tag and UMLS concepts. The sentence level features help the system by providing information regarding the context of the word including previous and next words and negations.

Using the extracted medical phrases as features, we address a classification task that classifies entire radiology reports as critical or non-critical (i.e. whether the emergency physician needs to take immediate action on them).

We also developed an active adaptive learning interface which bridges the gap of machine learning models and human feedback. This system provides the user with clinically significant phases extracted for each report and also helps the user in changing the predicted annotations based on the uncertainty level of predictions. This allows the doctors to easily identify the patient's condition and also helps the machine learning model to adjust the predictions of future reports as per the feedback. The interface provides visual cues to the user on the uncertainty level of the predicted phrases so that the user is aware of the level of confidence the system has on the predicted phrases. The interface also provides information on how certain it is on predicting the overall critical level of the report.

The research also contributes to the improved accuracy of word segmentation and error correction of radiology reports. The default word segmentation algorithms are based on the daily language spoken by the humans, which perform poorly on the complex medical documents. Our research was able to improve the accuracy of these algorithms considerably for medical text processing.

Our research was also able to identify and capture medical phrases used by emergency physicians and was also able to identify phrases which are considered most important during the diagnosis procedure. We were also able to capture medical phrases which are always considered as highly critical or critical to the patient as well.

5.1 Future Work

The current research focused on extracting clinically significant information and building a feature matrix using the output of the machine learning linear classifiers. We were able to segregate the clinically significant critical levels to 3 classes as high-critical, critical and non-critical values. And the simple active learning interface was able to provide the user with a visual interface which shows the various critical level phrases along with the confidence level. This active learning interface can be used for gathering further annotations for enhanced performance of the system.

An enhancement of this system can be implemented in high-level programming languages such as PHP and can be implemented as an end product for patient diagnosis. The model can be further tuned to process thousands of medical reports and rank the reports as per the overall criticality level which can help the doctors to prioritize patients based on criticality levels. Another use of the system is the patient history diagnosis in which the system is used to identify the critical medical information of a single patient over a period of time. This can help the doctors to pinpoint root causes of long-term medical conditions and can help in effective treatment of the patient.

Another enhancement of the system is by combining multiple types of reports of patients and identifying medical information or disease analysis. Combining the image data of chest X-rays and text radiology reports can provide the doctors with a plethora of information regarding the patient's condition. The system can be trained to identify disease patterns in the chest x-ray image and can process the radiology reports for descriptive details. This will require advanced machine learning object detection models for image processing combined with CRF or similar sequence learners for text processing. The feature matrix created by the CRF model in this research can also be used as additional feature vectors for advance machine learning algorithms for creating detailed patient profiling models or auto template mapping of patient records.

The active learning interface can also be extended to an advanced model in which all the records of the same patient are shown in a single interface along with the criticality terms extracted. The doctors could choose each phrase extracted and can visualize the different reports in which such phrase is present. This could also help in tagging multiple reports at the same time for gathering training data.

Further to this, the model itself can be changed to advance deep learning models (Collobert and Weston, 2008) . However, the current CRF model can help in gathering training data faster than manual methods. As more data is labeled using the CRF model, the labeled data can be used in deep networks for better performance. Another method to increase the number of labeled data is by using feedback approach mentioned in this research in Section4.8.

Bibliography

Aizawa, A. (2003). An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Amari, S.-i. (1993). Backpropagation and Stochastic Gradient Descent method. *Neurocomputing*, 5(4-5):185–196.

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Baldridge, J., Clark, P., and Tur, G. (2010). Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl 1):D267–D270.

Carreras, X., Màrquez, L., and Padró, L. (2003). A simple named entity extractor using AdaBoost. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 152–155. Association for Computational Linguistics.

Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., and Buchanan, B. G. (2001). A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics.

Collobert, R. and Weston, J. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 160–167, New York, NY, USA. ACM.

Daumé III, H. and Marcu, D. (2005). Learning as search optimization: Approximate large margin methods for structured prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 169–176. ACM.

Dreyer, K. (2014). Information theory entropy reduction program. US Patent 8,756,234.

Goldwater, S., Griffiths, T. L., and Johnson, M. (2009). A bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Gunn, S. R. et al. (1998). Support vector machines for classification and regression. *UNIVERSITY OF SOUTHAMPTON , Faculty of Engineering, Science and Mathematics, School of Electronics and Computer Science*, 14.

Halevy, A., Norvig, P., and Pereira, F. (2009). The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12.

Hall, F. M. (2000). Language of the radiology report: primer for residents and wayward radiologists. *American Journal of Roentgenology*, 175(5):1239–1242.

Hassanpour, S. and Langlotz, C. P. (2016). Information Extraction from Multi-institutional Radiology Reports. *Artificial intelligence in medicine*, 66:29–39.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Langlotz, C. P. (2006). Radlex: a new method for indexing online educational materials. *Radiographics*, 26(6):1595–1597.

Langlotz, C. P. and Meininger, L. (2000). Enhancing the expressiveness and usability of structured image reporting systems. In *Proceedings of the AMIA symposium*, page 467. American Medical Informatics Association.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomForest. *R news*, 2(3):18–22.

McCallum, A., Freitag, D., and Pereira, F. C. (2000). Maximum Entropy Markov Models for Information Extraction and Segmentation. In *ICML*, volume 17, pages 591–598.

Meystre, S. M., Thibault, J., Shen, S., Hurdle, J. F., and South, B. R. (2010). Textractor: a hybrid system for medications and reason for their prescription extraction from clinical text documents. *Journal of the American Medical Informatics Association*, 17(5):559–562.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Nocedal, J. (1980). Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782.

Patrick, J. and Li, M. (2009). A cascade approach to extracting medication events. In *Australasian Language Technology Association Workshop December-2009*, volume 7, page 99.

Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models. *IEEE assp magazine*, 3(1):4–16.

Refaeilzadeh, P., Tang, L., and Liu, H. (2009). Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer.

Segaran, T., Hammerbacher, J., and Norvig, P. (2009). *Natural language corpus data*, pages 219–242. O'Reilly Media.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 134–141. Association for Computational Linguistics.

Smith, L., Rindflesch, T., Wilbur, W. J., et al. (2004). Medpost: a part-of-speech tagger for bioMedical text. *Bioinformatics*, 20(14):2320–2321.

Sokolova, M., Japkowicz, N., and Szpakowicz, S. (2006). *Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation*, pages 1015–1021. Springer Berlin Heidelberg, Berlin, Heidelberg.

Sutton, C. and McCallum, A. (2010). An introduction to conditional random fields. *arXiv preprint arXiv:1011.4088*.

Xu, H., Stenner, S. P., Doan, S., Johnson, K. B., Waitman, L. R., and Denny, J. C. (2010). Medex: a medication information extraction system for clinical narratives. *Journal of the American Medical Informatics Association*, 17(1):19–24.

Yetisgen-Yildiz, M., Gunn, M. L., Xia, F., and Payne, T. H. (2013). A text processing pipeline to extract recommendations from radiology reports. *Journal of biomedical informatics*, 46(2):354–362.

Yim, W.-w., Denman, T., Kwan, S. W., and Yetisgen, M. (2016). Tumor information extraction in radiology reports for Hepatocellular Carcinoma patients. *AMIA Summits on Translational Science Proceedings*, 2016:455.

Zhang, S., Wu, Y., Che, T., Lin, Z., Memisevic, R., Salakhutdinov, R., and Bengio, Y. (2016). Architectural Complexity Measures of Recurrent Neural Networks. *arXiv preprint arXiv:1602.08210*.

Appendix A

Training CRF model

This section shows the experimental results for training the CRF model in detail.

A.1 Cross validation performance across folds for CRF model

In our thesis the CRF model is trained by using 10 fold cross validation. The samples for training and testing are selected by random sampling with replacement. The following tables show the various performance scores achieved on training the model based on random sampling. The sampling of the data is done at sentence level which provides a good balance between training and testing data sets across each folds. If the sampling is done at report level, the imbalance of training testing data would be large. This is because of the inconsistency in the report length(number of sentences in each report).

Performance of the model is measured based on the precision, recall and f1-score of the predicted labels.The best average performance was obtained on fold7 with an average f1-score of 79.52 and the worst f1-score was obtained on the fold4 with average f1-score as 68.52.

Table A.1: Performance scores for CRF model for Fold1. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.7821	0.6932	0.7349
I-NonCrit	0.7816	0.8095	0.7953
B-Crit	0.7708	0.7708	0.7708
I-Crit	0.9762	0.8723	0.9213
B-HighCrit	0.6667	0.6667	0.6667
I-HighCrit	0.8261	0.8837	0.8539
avg / total	0.7947	0.7729	0.7826

Table A.2: Performance scores for CRF model for Fold2. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.7917	0.6552	0.717
I-NonCrit	0.8529	0.7699	0.8093
B-Crit	0.878	0.6102	0.72
I-Crit	0.6591	0.7073	0.6824
B-HighCrit	0.68	0.7556	0.7158
I-HighCrit	0.6122	0.9375	0.7407
avg / total	0.7814	0.7192	0.7414

Table A.3: Performance scores for CRF model for Fold3. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.7222	0.6989	0.7104
I-NonCrit	0.747	0.8378	0.7898
B-Crit	0.6977	0.75	0.7229
I-Crit	0.5556	0.75	0.6383
B-HighCrit	0.8772	0.8197	0.8475
I-HighCrit	0.8444	0.95	0.8941
avg / total	0.7584	0.7927	0.7733

Table A.4: Performance scores for CRF model for Fold4. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.7595	0.5714	0.6522
I-NonCrit	0.6848	0.7326	0.7079
B-Crit	0.6792	0.6207	0.6486
I-Crit	0.6667	0.5854	0.6234
B-HighCrit	0.6786	0.7451	0.7103
I-HighCrit	0.6949	0.9318	0.7961
avg / total	0.7027	0.6805	0.6852

Table A.5: Performance scores for CRF model for Fold5. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.7714	0.6328	0.6953
I-NonCrit	0.8172	0.717	0.7638
B-Crit	0.6061	0.5405	0.5714
I-Crit	0.5833	0.5833	0.5833
B-HighCrit	0.6889	0.7561	0.7209
I-HighCrit	0.7273	0.8276	0.7742
avg / total	0.7428	0.674	0.7044

Table A.6: Performance scores for CRF model for Fold6. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.6966	0.7045	0.7006
I-NonCrit	0.7386	0.8228	0.7784
B-Crit	0.8983	0.7361	0.8092
I-Crit	0.7941	0.8438	0.8182
B-HighCrit	0.8065	0.8929	0.8475
I-HighCrit	0.8846	0.9583	0.92
avg / total	0.785	0.7895	0.7845

Table A.7: Performance scores for CRF model for Fold7. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.8434	0.7368	0.7865
I-NonCrit	0.8452	0.8659	0.8554
B-Crit	0.6977	0.7692	0.7317
I-Crit	0.7045	0.8857	0.7848
B-HighCrit	0.86	0.7049	0.7748
I-HighCrit	0.8378	0.7561	0.7949
avg / total	0.8162	0.7819	0.7952

Table A.8: Performance scores for CRF model for Fold8. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.8333	0.8065	0.8197
I-NonCrit	0.8471	0.878	0.8623
B-Crit	0.6939	0.6296	0.6602
I-Crit	0.6444	0.7436	0.6905
B-HighCrit	0.7241	0.7636	0.7434
I-HighCrit	0.717	0.76	0.7379
avg / total	0.7647	0.7775	0.7702

Table A.9: Performance scores for CRF model for Fold9. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.8315	0.7048	0.7629
I-NonCrit	0.8608	0.85	0.8553
B-Crit	0.6531	0.6667	0.6598
I-Crit	0.8438	0.6136	0.7105
B-HighCrit	0.8333	0.7447	0.7865
I-HighCrit	0.8286	0.8286	0.8286
avg / total	0.8156	0.7382	0.7728

Table A.10: Performance scores for CRF model for Fold10. Training and testing data is obtained based on random sampling at sentence level.

	precision	**recall**	**f1-score**
B-NonCrit	0.7778	0.8116	0.7943
I-NonCrit	0.8611	0.8732	0.8671
B-Crit	0.625	0.5682	0.5952
I-Crit	0.6957	0.7273	0.7111
B-HighCrit	0.8056	0.7733	0.7891
I-HighCrit	0.9091	0.6667	0.7692
avg / total	0.7943	0.7577	0.773

Below tables shows the detailed confusion matrix for the various folds of CRF model.

Table A.11: Confusion Matrix for fold1 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	58	5	0	0	1	0	19
	I-NonCrit	2	61	0	0	0	1	12
	B-Crit	1	0	42	1	8	1	12
	I-Crit	0	2	2	27	0	4	7
	B-HighCrit	0	0	1	0	40	0	1
	I-HighCrit	0	0	0	0	1	36	1
	O	9	9	4	3	0	2	1379

Table A.12: Confusion Matrix for fold2 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	66	6	2	0	2	0	31
	I-NonCrit	2	60	3	1	1	3	14
	B-Crit	0	0	36	1	5	1	8
	I-Crit	0	0	1	20	0	3	2
	B-HighCrit	2	0	4	0	39	1	3
	I-HighCrit	0	1	1	0	3	31	2
	O	10	8	5	7	2	1	1361

Table A.13: Confusion Matrix for fold3 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	75	4	4	1	4	0	31
	I-NonCrit	1	77	1	3	0	2	7
	B-Crit	0	0	32	0	3	0	7
	I-Crit	0	0	2	24	0	1	3
	B-HighCrit	1	0	4	1	38	2	3
	I-HighCrit	0	1	0	2	2	25	7
	O	14	13	6	8	2	3	1299

Table A.14: Confusion Matrix for fold4 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	64	3	3	2	0	0	17
	I-NonCrit	0	54	1	4	0	0	12
	B-Crit	2	0	39	1	5	2	13
	I-Crit	0	2	1	42	0	3	11
	B-HighCrit	5	0	5	0	36	1	2
	I-HighCrit	0	0	0	0	0	23	0
	O	13	11	2	5	6	4	1333

Table A.15: Confusion Matrix for fold5 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	60	1	1	0	1	0	24
	I-NonCrit	2	62	0	2	0	1	12
	B-Crit	2	0	35	1	2	1	3
	I-Crit	0	2	1	38	0	2	1
	B-HighCrit	6	0	5	0	41	4	0
	I-HighCrit	0	2	0	0	2	31	2
	O	7	11	4	1	3	2	1405

Table A.16: Confusion Matrix for fold6 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	52	4	3	0	2	0	24
	I-NonCrit	3	55	0	1	1	0	7
	B-Crit	4	0	50	0	9	2	12
	I-Crit	0	1	0	43	0	4	11
	B-HighCrit	1	0	2	1	35	2	2
	I-HighCrit	0	1	0	2	1	36	0
	O	8	9	6	8	2	3	1396

Table A.17: Confusion Matrix for fold7 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	77	5	2	0	3	0	12
	I-NonCrit	1	65	0	1	0	1	3
	B-Crit	1	0	27	2	3	1	4
	I-Crit	0	0	1	15	0	0	1
	B-HighCrit	0	0	4	0	40	0	1
	I-HighCrit	0	0	0	0	1	30	0
	O	18	19	5	5	3	2	1218

Table A.18: Confusion Matrix for fold8 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	64	6	3	1	3	0	14
	I-NonCrit	1	58	1	2	1	1	8
	B-Crit	3	0	20	2	5	0	9
	I-Crit	0	4	0	17	0	3	3
	B-HighCrit	2	0	5	0	47	2	7
	I-HighCrit	0	0	2	2	1	38	7
	O	14	6	6	4	3	0	1389

Table A.19: Confusion Matrix for fold9 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	57	5	1	0	2	0	24
	I-NonCrit	1	56	1	0	0	1	8
	B-Crit	3	0	36	1	7	1	3
	I-Crit	0	0	0	25	0	5	0
	B-HighCrit	2	0	6	0	42	3	2
	I-HighCrit	0	0	0	1	1	37	0
	O	14	18	4	5	2	3	1365

Table A.20: Confusion Matrix for fold10 of the CRF model training.

		Predicted						
		B-NonCrit	I-NonCrit	B-Crit	I-Crit	B-HighCrit	I-HighCrit	O
Actual	B-NonCrit	61	5	1	1	4	0	18
	I-NonCrit	1	58	2	1	1	3	8
	B-Crit	3	0	31	1	3	0	6
	I-Crit	0	2	0	21	0	0	0
	B-HighCrit	1	1	5	0	46	2	1
	I-HighCrit	0	2	0	2	2	31	3
	O	16	10	7	5	5	3	1399

Appendix B

Structured Perceptron Performance

Table B.1: Performance of Structured Perceptron for fold 1. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.7683	0.6176	0.6848
I-NonCrit	0.8072	0.7204	0.7614
B-Crit	0.6167	0.6981	0.6549
I-Crit	0.6111	0.5641	0.5867
B-HighCrit	0.6852	0.6066	0.6435
I-HighCrit	0.6939	0.7083	0.701
avg / total	0.7198	0.6566	0.6847

Table B.2: Performance of Structured Perceptron for fold 2. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.7143	0.6742	0.6936
I-NonCrit	0.7213	0.6377	0.6769
B-Crit	0.5476	0.5897	0.5679
I-Crit	0.4737	0.72	0.5714
B-HighCrit	0.7174	0.6111	0.66
I-HighCrit	0.7241	0.6176	0.6667
avg / total	0.6771	0.6419	0.6554

Table B.3: Performance of Structured Perceptron for fold 3. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.7812	0.8152	0.7979
I-NonCrit	0.8571	0.8276	0.8421
B-Crit	0.7317	0.6667	0.6977
I-Crit	0.7188	0.7667	0.7419
B-HighCrit	0.7593	0.8367	0.7961
I-HighCrit	0.8293	0.8947	0.8608
avg / total	0.7908	0.8065	0.7978

Table B.4: Performance of Structured Perceptron for fold 4. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.75	0.7429	0.7464
I-NonCrit	0.7755	0.8352	0.8042
B-Crit	0.65	0.6842	0.6667
I-Crit	0.5333	0.64	0.5818
B-HighCrit	0.7105	0.6585	0.6835
I-HighCrit	0.9032	0.7368	0.8116
avg / total	0.742	0.7426	0.7405

Table B.5: Performance of Structured Perceptron for fold 5. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.7949	0.6458	0.7126
I-NonCrit	0.7671	0.7179	0.7417
B-Crit	0.661	0.629	0.6446
I-Crit	0.7174	0.6875	0.7021
B-HighCrit	0.6596	0.6739	0.6667
I-HighCrit	0.7105	0.75	0.7297
avg / total	0.7308	0.6776	0.7018

Table B.6: Performance of Structured Perceptron for fold 6. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.7952	0.6667	0.7253
I-NonCrit	0.7808	0.7308	0.755
B-Crit	0.7083	0.6939	0.701
I-Crit	0.8519	0.6389	0.7302
B-HighCrit	0.7018	0.7843	0.7407
I-HighCrit	0.7143	0.8824	0.7895
avg / total	0.7639	0.7205	0.7376

Table B.7: Performance of Structured Perceptron for fold 7. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.72	0.6279	0.6708
I-NonCrit	0.8732	0.9254	0.8986
B-Crit	0.6667	0.5652	0.6118
I-Crit	0.5882	0.8	0.678
B-HighCrit	0.7344	0.7231	0.7287
I-HighCrit	0.7778	0.7568	0.7671
avg / total	0.7433	0.727	0.7323

Table B.8: Performance of Structured Perceptron for fold 8. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.6966	0.6739	0.6851
I-NonCrit	0.8485	0.7	0.7671
B-Crit	0.8305	0.6806	0.7481
I-Crit	0.8158	0.6596	0.7294
B-HighCrit	0.8333	0.7609	0.7955
I-HighCrit	0.7931	0.8519	0.8214
avg / total	0.7963	0.7033	0.7454

Table B.9: Performance of Structured Perceptron for fold 9. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.7037	0.76	0.7308
I-NonCrit	0.7612	0.7846	0.7727
B-Crit	0.75	0.65	0.6964
I-Crit	0.7	0.5385	0.6087
B-HighCrit	0.6774	0.9333	0.785
I-HighCrit	0.6809	0.8889	0.7711
avg / total	0.7173	0.7562	0.7301

Table B.10: Performance of Structured Perceptron for fold 10. Training and testing data is random sampling with replacement.

	precision	**recall**	**f1-score**
B-NonCrit	0.6932	0.7011	0.6971
I-NonCrit	0.759	0.6702	0.7119
B-Crit	0.7302	0.7797	0.7541
I-Crit	0.7632	0.6444	0.6988
B-HighCrit	0.6739	0.775	0.7209
I-HighCrit	0.5319	0.9615	0.6849
avg / total	0.7119	0.7265	0.7127

Appendix C

Two Class system for CRF model

In the two class model, the criticality levels are reduced from three levels to two. High-critical and critical level phrases are combined into one critical level. Such reduction in critical level value increased the accuracy of the model and generalized the model further. The performance of the two class CRF model and the confusion matrix of such model is given in below tables.

Table C.1: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 1

	precision	**recall**	**f1-score**
B-NonCrit	0.8194	0.7108	0.7613
I-NonCrit	0.7922	0.8026	0.7974
B-Crit	0.93	0.8692	0.8986
I-Crit	0.9067	0.85	0.8774
avg / total	0.8678	0.8121	0.8385

Table C.2: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 2

	precision	**recall**	**f1-score**
B-NonCrit	0.85	0.6355	0.7273
I-NonCrit	0.8158	0.7381	0.775
B-Crit	0.7885	0.82	0.8039
I-Crit	0.7397	0.8438	0.7883
avg / total	0.8047	0.7493	0.7712

Table C.3: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 3

	precision	**recall**	**f1-score**
B-NonCrit	0.8261	0.6387	0.7204
I-NonCrit	0.8125	0.8571	0.8342
B-Crit	0.7879	0.8571	0.8211
I-Crit	0.7049	0.6418	0.6719
avg / total	0.7912	0.7473	0.7646

Table C.4: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 4

	precision	**recall**	**f1-score**
B-NonCrit	0.7805	0.7191	0.7485
I-NonCrit	0.7941	0.7606	0.777
B-Crit	0.8654	0.8108	0.8372
I-Crit	0.7412	0.7683	0.7545
avg / total	0.8008	0.7677	0.7835

Table C.5: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 5

	precision	**recall**	**f1-score**
B-NonCrit	0.686	0.6782	0.6821
I-NonCrit	0.7531	0.7722	0.7625
B-Crit	0.8416	0.85	0.8458
I-Crit	0.814	0.8642	0.8383
avg / total	0.776	0.7925	0.784

Table C.6: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 6

	precision	**recall**	**f1-score**
B-NonCrit	0.7286	0.6	0.6581
I-NonCrit	0.7671	0.8358	0.8
B-Crit	0.8649	0.8	0.8312
I-Crit	0.85	0.8586	0.8543
avg / total	0.812	0.7763	0.792

Table C.7: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 7

	precision	**recall**	**f1-score**
B-NonCrit	0.8039	0.8283	0.8159
I-NonCrit	0.7444	0.9437	0.8323
B-Crit	0.8778	0.9518	0.9133
I-Crit	0.8491	0.9375	0.8911
avg / total	0.8175	0.907	0.8586

Table C.8: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 8

	precision	**recall**	**f1-score**
B-NonCrit	0.7683	0.6923	0.7283
I-NonCrit	0.8169	0.8056	0.8112
B-Crit	0.7692	0.7843	0.7767
I-Crit	0.8289	0.8182	0.8235
avg / total	0.7925	0.7719	0.7816

Table C.9: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 9

	precision	**recall**	**f1-score**
B-NonCrit	0.7368	0.6292	0.6788
I-NonCrit	0.6835	0.806	0.7397
B-Crit	0.8692	0.8774	0.8732
I-Crit	0.7528	0.971	0.8481
avg / total	0.7718	0.8157	0.7887

Table C.10: Performance of the CRF two class model with critical and non-critical phrases extracted for fold 10

	precision	**recall**	**f1-score**
B-NonCrit	0.759	0.7	0.7283
I-NonCrit	0.7632	0.7838	0.7733
B-Crit	0.7876	0.89	0.8357
I-Crit	0.6591	0.9206	0.7682
avg / total	0.7495	0.8196	0.779

Table C.11: Confusion matrix for two class CRF model for fold 1

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	59	4	1	1	18
	I-NonCrit	2	61	0	1	12
	B-Crit	2	0	93	1	11
	I-Crit	0	3	3	65	9
	O	9	9	4	2	1382

Table C.12: Confusion matrix for two class CRF model for fold 2

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	68	6	6	0	27
	I-NonCrit	1	62	3	6	12
	B-Crit	2	0	81	5	12
	I-Crit	0	1	5	52	6
	O	9	7	9	8	1361

Table C.13: Confusion matrix for two class CRF model for fold 3

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	75	4	8	1	31
	I-NonCrit	1	77	1	5	7
	B-Crit	1	0	79	2	9
	I-Crit	0	1	4	43	19
	O	14	13	8	10	1300

Table C.14: Confusion matrix for two class CRF model for fold 4

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	62	3	3	2	19
	I-NonCrit	0	54	1	4	12
	B-Crit	4	0	90	6	11
	I-Crit	0	0	1	63	18
	O	15	10	9	10	1330

Table C.15: Confusion matrix for two class CRF model for fold 5

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	59	3	5	1	19
	I-NonCrit	2	61	0	5	11
	B-Crit	9	0	84	5	2
	I-Crit	0	6	3	69	3
	O	15	12	7	6	1393

Table C.16: Confusion matrix for two class CRF model for fold 6

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	51	4	4	0	26
	I-NonCrit	4	55	1	1	6
	B-Crit	7	0	97	4	12
	I-Crit	0	4	1	85	9
	O	8	9	9	10	1396

Table C.17: Confusion matrix for two class CRF model for fold 7

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	82	4	4	0	9
	I-NonCrit	1	67	0	1	2
	B-Crit	1	0	79	1	2
	I-Crit	0	0	2	45	1
	O	18	19	5	6	1222

Table C.18: Confusion matrix for two class CRF model for fold 8

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	63	6	8	1	13
	I-NonCrit	1	58	2	4	7
	B-Crit	4	0	82	4	12
	I-Crit	0	1	2	65	9
	O	14	6	11	4	1387

Table C.19: Confusion matrix for two class CRF model for fold 9

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	57	4	3	0	25
	I-NonCrit	1	56	1	1	8
	B-Crit	4	0	93	5	4
	I-Crit	0	1	1	67	0
	O	15	18	9	16	1353

Table C.20: Confusion matrix for two class CRF model for fold 10

		Predicted				
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**O**
Actual	**B-NonCrit**	63	5	5	1	16
	I-NonCrit	1	58	3	4	8
	B-Crit	3	1	87	3	6
	I-Crit	0	2	2	56	3
	O	18	12	14	22	1379

Appendix D

Feature Weights on CRF model

This section shows the detailed positive and negative feature weight values for each criticality levels on training the CRF model.

Table D.1: Top ten positive features for CRF model. The higher the weight, the higher the significance of the feature in deciding the high-critical class.

B-HIGH CRIT		**I-HIGH CRIT**	
Feature	**Weight**	**Feature**	**Weight**
'highFlag:True'	1.83799	'word2VecSimilarityPrev:0.999597393253'	1.500624
'nextWord1:can'	1.775523	'nextPos:superior ,'	1.458598
'prevWord1:lobe'	1.540945	'highFlag:True'	1.41717
'suffix1:is'	1.485263	'word2VecSimilarityPrev:0.999418976773'	1.320887
'prevWord1:a'	1.464373	'suffix1:in'	1.112949
'prevWord1:breath'	1.379365	'prevWordPos1:prep'	0.830228
'negExPrev:'	1.320104	'negExPrev:'	0.7017
'opacity'	1.279762	'prevWord1:chest'	0.671723
metaConcept:'Sign or Symptom', 'Pathologic Function'	1.229902	'word2VecSimilarityPrev:0.99961038638'	0.633416
'suffix1:us'	1.023736	'prevWordPos1:det'	0.620302

Table D.2: Top ten negative features for CRF model. The lower the weight, the higher the significance of the feature in deciding against the high-critical class.

B-HIGH CRIT		I-HIGH CRIT	
Feature	**Weight**	**Feature**	**Weight**
'nextWordPos1:adj'	-0.489813	'pos_tag:adj'	-0.41624
'prevWordPos1:det'	-0.491402	'nextWord1:at'	-0.45768
'pref2:bre'	-0.49752	'pos_tag:verb'	-0.47826
'critFlag:True'	-0.823967	'pref1:le'	-0.50878
'nonCritFlag:True'	-0.934061	metaConcept:'Qualitative Concept'	-0.56377
'prevWordPos1:adj'	-0.955369	'prevWord1:disease'	-0.59948
'nextNeg:chronic '	-1.120375	'nextWordPos1:noun'	-0.73814
metaConcept:'Spatial Concept'	-1.132527	'pref1:me'	-1.08589
'isHIghCrit:'	-1.930719	'highFlag:'	-1.67584
'highFlag:'	-2.28174	'isHIghCrit:'	-1.84714

Table D.3: Top ten positive features for CRF model. The higher the weight, the higher the significance of the feature in deciding the critical class.

B-Crit		I-Crit	
Feature	**Weight**	**Feature**	**Weight**
'nextNeg:chronic '	1.537657	'critFlag:True'	1.868761
'prevWord1:mild'	1.185485	'pos_tag:noun'	1.27514
metaConcept:'Pathologic Function', 'Finding'	1.174125	'nextWord1:unchanged'	1.135037
'nextWord1:areas'	1.143191	'word2VecSimilarityPrev:0.999558585312'	0.977696
'prevWord1:known'	1.14195	'word2VecSimilarityPrev:0.999675685209'	0.966688
'critFlag:True'	1.136861	'nextWord1:has'	0.77849
'nextWordPos1:modal'	1.081014	'nextWord1:most'	0.652714
'prevWord1:Mild'	1.053178	'nextWord1:in'	0.574062
'nextWord1:thickening'	1.04016	'word2VecSimilarityPrev:0.999549075206'	0.571464
'prevWord1:The'	0.972441	'silhouette'	0.559219

Table D.4: Top ten negative features for CRF model. The lower the weight, the higher the significance of the feature in deciding against the critical class.

B-Crit		**I-Crit**	
Feature	**Weight**	**Feature**	**Weight**
'pref1:ch'	-0.49806	'pos_tag:adj'	-0.39164
'word2VecSimilarityPrev:0.999468437275'	-0.52439	'negExNext:'	-0.45231
'pref2:con'	-0.53231	'pref1:lu'	-0.46868
'nextWord1:and'	-0.7035	'suffix1:al'	-0.50963
'nonCritFlag:True'	-0.72214	'prevWord1:unfolding'	-0.57372
'prevWord1:increased'	-0.73806	'nonCritFlag:True'	-0.60042
'suffix1:le'	-0.87009	'pref1:pr'	-0.60831
'nextWord1:lung'	-0.96504	metaConcept:'Spatial Concept'	-0.73183
'critFlag:'	-1.53158	'isCrit:'	-0.91812
'isCrit:'	-2.73757	'critFlag:'	-1.74695

Table D.5: Top ten positive features for CRF model. The higher the weight, the higher the significance of the feature in deciding the non-critical class.

B-NON CRIT		**I-NON CRIT**	
Feature	**Weight**	**Feature**	**Weight**
'word2VecSimilarityPrev:0.999108931789'	1.754375	'nextPos:improvement ,'	1.441279
'negExPrev:1'	1.691163	'nextWord1:pattern'	1.38209
'prevWord1:consolidation'	1.635556	'nextWord1:base'	1.313811
'nextPos:appreciated ,'	1.623188	'nonCritFlag:True'	1.106649
'prevWord1:based'	1.546116	'nextNeg:no no ,'	1.059263
'nextWord1:or'	1.5279	'prevWord1:well'	1.046057
'prevWord1:prior'	1.481233	'word2VecSimilarityPrev:0.999179613068'	1.015431
'prevWord1:definite'	1.448797	'nextWord1:are'	1.010685
'prevWord1:or'	1.299454	'prevWord1:chest'	1.005939
'prevWord1:No'	1.254985	'pref1:ch'	0.915598

Table D.6: Top ten negative features for CRF model. The lower the weight, the higher the significance of the feature in deciding against the non-critical class.

B-NON CRIT		**I-NON CRIT**	
Feature	**Weight**	**Feature**	**Weight**
'prevWord1:a'	-0.715994	'word2VecSimilarityPrev:0.998901530601'	-0.548026
'suffix1:ly'	-0.763041	'nextWord1:mediastinum'	-0.61556
'negExPrev:15'	-0.766513	'pref1:ma'	-0.69287
'pref2:int'	-0.802672	'suffix1:um'	-0.717939
'suffix1:er'	-0.847081	'nextWord1:effusion'	-0.721596
'word2VecSimilarityPrev:0.999610326313'	-0.933249	'nextWord1:has'	-0.738964
'prevWordPos1:adj'	-1.007836	'nextWordPos1:prep'	-0.954095
'prevWord1:with'	-1.074366	'nonCritFlag:'	-0.977529
'suffix1:ma'	-1.145215	'prevWord1:mediastinal'	-1.365657
'isNonCrit:'	-1.424129	'isNonCrit:'	-1.372418

Appendix E

Radiology CRF model trained on abdominal dataset

To check the effectiveness of the features and the algorithm, we trained the CRF model with same features on top of another dataset. The new dataset was abdominal data de-identified of patient information and spell and error corrected using our algorithm. The training data was smaller than the number of reports we trained for the chest data set. We trained the abdominal dataset with 104 reports tagged by the physician using our tagging interface.

Even though the number of training data was considerably less than the chest X-ray reports, the model was able to perform well and we were able to obtain an average f1-score of 0.71 with an impressive precision of 0.75. The performance scores of he abdominal dataset trained using CRF model is shown in Tables E.1 E.2 below.

Table E.1: performance of the CRF model trained on 104 abdominal radiology reports. Performance is measured using ten fold cross validation.

	precision	**recall**	**f1-score**
B-NonCrit	0.79378	0.74487	0.76779
I-NonCrit	0.75198	0.73469	0.73851
B-Crit	0.66284	0.55156	0.601
I-Crit	0.7299	0.6533	0.6877
B-HighCrit	0.78317	0.72231	0.74808
I-HighCrit	0.76652	0.68206	0.70839
avg / total	0.75403	0.68609	0.71372

Table E.2: Confusion matrix for CRF model trained on abdominal dataset.

		Predicted						
		B-NonCrit	**I-NonCrit**	**B-Crit**	**I-Crit**	**B-HighCrit**	**I-HighCrit**	**O**
Actual	**B-NonCrit**	49	4	1	0	1	0	11
	I-NonCrit	3	95	0	2	0	2	30
	B-Crit	1	0	22	4	2	0	10
	I-Crit	0	1	4	75	0	4	32
	B-HighCrit	1	0	1	0	26	2	6
	I-HighCrit	0	2	0	1	2	51	21
	O	8	26	5	20	2	8	1369

Appendix F

Software tools and Packages used

Table F.1: Python packages used for this research.

Package	Module	Description
nltk.corpus	stopwords	Used for removing stop words
nltk.stem.porter	PorterStemmer	For stemming a given word
pickle	-	Storing python objects as dat files. Useful for storing tagged data and for saving feature matrix.
csv	-	For reading csv formated files, Radiology datasets stored in csv format by default.
subprocess	-	Executing unix command line arguments from python program. Useful for executing MetaMap tool from the python program.
re	-	regular expression processing.
nltk.stem	WordNetLemmatizer	Lemmatize given word
sklearn.metrics	classification_report, confusion_matrix	Creating confusion matrix for the model
random	-	Randomized sample selection
sklearn_crfsuite	-	crf suite library
sklearn_crfsuite	scorers,metrics	For calculating performance metrics
sklearn.grid_search	RandomizedSearchCV	optimizing parameter set for the model
seqlearn.perceptron	StructuredPerceptron	Structured Perceptron model implementation
sklearn	cross_validation	cross validation package
os	-	file operations
cgi,cgitg	-	javascript to python value passing
sklearn.linear_model	SGDClassifier	SGD Classifier for report classification
sklearn.feature_extraction.text	CountVectorizer,TfidfTransformer	TF idf score generation
sklearn.svm	LinearSVC	Linear SVM implementation
sklearn.ensemble	RandomForestClassifier	RF classifier
sklearn.datasets	load_svmlight_file	loading files to dataset
sklearn	preprocessing	Preprocessing of dataset values

Some of the other external tools used on this research are:

- MedPost SKR Tagger - This is an external Part of Speech tagging tool developed for tagging medical text data. The full version and details can be found at https://metamap.nlm.nih.gov/MedPostSKRTagger.shtml
- MetaMap - Another tool used in this research is the MetaMap. This tool provides the UMLS tags and concept of a given word. The tool is installed as a local instance. Complete details can be found at https://metamap.nlm.nih.gov/

Appendix G

Copyright Notice