Syntactic NLP Techniques on Natural Language in Clinical Decision Support Engine

Prepared By :

Mayank Gour

12MICT07

Internal Guide	***External Guide***
Dr. Sanjay Garg	**Mr. Satish Vagadia**
Nirma University	**J-KRI TechLabs, Pune.**

Department Of Computer Science And Engineering

Institute Of Technology

Nirma University

Ahmedabad

May-2014

Syntactic NLP Techniques on Natural Language in Clinical Decision Support Engine

Major Project

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

PREPARED BY :

Mayank Gour

12MICT07

Internal Guide
Dr. Sanjay Garg
Nirma University

External Guide
Mr. Satish Vagadia
J-KRI TechLabs, Pune.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD

CERTIFICATE

This is to certify that the Major Project entitled **Syntactic NLP Techniques on Natural Language in Clinical Decision Support Engine** submitted by **Mayank Gour (12MICT07)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science Engineering of Nirma University of Science and Technology, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project, to the best of my knowledge, have not been submitted to any other university or institution for award of any degree or diploma.

Mr. Satish Vagadia
External Guide,
J-KRI TechLabs, Pune.

Dr. Sanjay Garg
Internal Guide & HOD-CSE,
Nirma University

Prof. Gaurang Raval
PG Coordinator - CSE(NT),
Nirma University

Dr. Ketan Kotecha
Director,
Nirma University

DECLARATION

This is to certify that,

I, **Mayank Gour**, **12MICT07**, a student of semester III Master of Technology in Computer Science Engineering, Nirma University, Ahmedabad, hereby declare that the project work **Syntactic NLP Techniques on Natural Language in Clinical Decision Support Engine** has been carried out by me under the guidance of Mr. Satish Vagadia, J-KRI TechLabs, Pune and Dr. Sanjay Garg, Department of Computer Science and Engineering, Nirma University, Ahmedabad. This Project has been submitted in the partial fulfillment of the requirements for the award of degree Master of Technology (M.Tech.) in Computer Science and Engineering (Networking Technology), Nirma University, Ahmedabad during the year 2013 - 2014.

I have not submitted this work in full or part to any other University or Institution for the award of any other degree.

Mayank Gour (12MICT07)

ACKNOWLEDGEMENT

I would like to thank my Mentor, **Mr. Satish Vagadia**, J-KRI TechLabs, Pune for his valuable guidance. Throughout the training, he has given me much valuable advice on project work. Without him, this project work would never have been completed.

I would also like to thank my Internal guide **Dr. Sanjay Garg**, Institute of Technology, Nirma University, Ahmedabad for her valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

I would also like to thank **Dr. Viral Bhatt**, J-KRI TechLabs, Pune for providing me all the information related to Medical Domain thought-out the duration of the project.

I would also like to thank **Dr. K.R.Kotecha**, Director, Institute of Technology, Nirma University, Ahmedabad for providing me an opportunity to get an internship at J-KRI TechLabs, Pune.

I would like to thank my all faculty members for providing encouragement, exchanging knowledge during my post-graduate program.

The blessings of God and family members make the way for completion of Project. I am very much grateful to them.

Mayank Gour (12MICT07)

Abstract

The primary intention of the project is to build a clinical decision support engine, a system for assisting users in their medical emergency in making medical decisions in scenarios with constraints on time and knowledge. The engine initially looks similar to a search engine which facilitate users to search medical remedies. Resultant system will produce the immediate response to the user query and will be crucial to improve Golden Period of the user in emergency. System will also provide information about top five nearest hospitals/laboratories/pharmacy depending on the users need including ambulance services.

Here the focus is on the use of NLP techniques, which later combines with Classification techniques to provide precise denition of entities and their relationships in order to generate native sub-query, which will work as input for other Engines. System will also try to extract users approximate location with several other parameters using their connection information which will be useful in various means.

Contents

Certificate iii

Declaration iv

Acknowledgement v

Abstract vi

List of Figures ix

1 Introduction 1

1.1 Scope of Work 2

2 Motivation 4

3 Information Retrieval and Natural Language Processing 6

3.1 Information Retrieval 6

3.2 Natural Language Processing 7

3.2.1 Syntactic NLP Techniques 8

4 Related Research 11

4.1 Stanford CoreNLP 11

4.2 WordNet 12

4.3 Natural Language Toolkit(NLTK) 12

4.4 PhraseNet 13

4.5 OpenNLP 13

4.6 Gensim 13

4.7 General Architecture for Text Engineering (GATE) 14

5 Language Modeling 15

5.1 Naive Bayes classifiers 15

5.2 Maximum Entropy Classifiers 16

5.3 Generative vs Conditional Classifiers 17

6 The Proposed Algorithm 20

6.1 Phase I: Appling NLP Techniques with Network Processing 20

6.2 Phase II: Classification Algorithm for Naive Bayesian method 22

6.3 How It Works 23

7 Implementation and Results 24

8 Conclusion and Future Work 31

List of Figures

2.1 Golden Hour Principle 4

3.1 First step of Basic IR process 7

3.2 Syntactic Parse Tags 9

5.1 An illustration of Naive Bayes procedure 16

5.2 Disease Category 19

6.1 Proposed Algorithm 21

6.2 Working of NB Algorithm 23

7.1 Home Page 28

7.2 Facility Registration: (Blank Form) 29

7.3 Facility Registration: (Filled Form) 30

Chapter 1

Introduction

The complexities of human language make searching for information encoded in natural language far from simple. The meaning, whether subtle or obvious to the reader, may be implicit in the language. Keyword search limits users to locating information specified explicitly at the surface level. Current Natural Language Processing (NLP) provisions perform reckonings over extensive corpora. With expanding recurrence, NLP provisions utilize the Web as their corpus and depend on queries to business search engines to backing these processing. Thus, the provisions are compelled to issue truly a huge number of queries to search engines, which can over-burden search engines, and made the application to perform slower.

In case of medical emergency, the existing system are manual system. Here the user have to manually ask for the information about hospitals, Medical store, Radiology etc. and have to refer that information. User can take help from doctor or other people. So what happens it will take more time and it will dangerous for the patients health condition. Our system will perform vital role in the golden period of the person who meets an accident. By using this system he can approach the nearest medical emergency service center in short period of time.

It also suggest medical remedies to the user who queried for it. In this Engine user will search for one liner query which will be in the form of Natural Language. System will produce the immediate response to the user query and will be crucial to improve Golden Period of the user in emergency. Medical search engine will search the respective needed medical center may be Hospitals, Blood Banks etc., by locating IP addresses or GPRS locator of the user. Different places can be identified by using respective pin code or zip code also system will make the use of longitude & latitude of geographical location.

For searching required medical center system will get the result as:

- First preference is to be given those facilities which are from GOVT, NGOs TRUSTs means low cost and public oriented services.
- On typing any condition, quick remedy with list of required facilities, hierarchy of selected facilities in relative distance from user, overall feedback.
- Facility display in search result: distance from user, contact number, local helpline.
- Facility display on click of facility: name of facility, relative distance, associated 24x7 services, contact no, feedback (short & long)

Information Verification & Feedback mechanism:

- User can give feedback for evaluation of service.
- On receiving new facility registration, system will send verification prompt to nearby facilities like: Do you know this List of facilities with check box of yes no no idea.
- system will also store that Who has replied yes no any idea to Whom means which facility has replied what answer to which facility. This will build referral relations for patients.
- All the information provided by our servers will be from authentic sources only. If not it will display a tag like This information is not authentic.

1.1 Scope of Work

The scope of this project includes generation of the Native Sub-Query from the user query with required parameters. End query should probably have following parameters:

1. Location Parameters:

    a. City

    b. Latitude

    c. Longitude

    d. Time

    e. Date

2. Other Parameters:

   a. Disease Category

   b. Gender

   c. Age

   d. Disease

   e. User Query

Chapter 2

Motivation

Concerning emergency trauma care, a couple of minutes can mean the distinction between life and demise. This first hour of definitive medical care is known as the "golden hour". It is normally this first hour where the patient's medical destiny is fixed. when all is said in done, the quicker that medical care is rendered, the better the medical result will be.

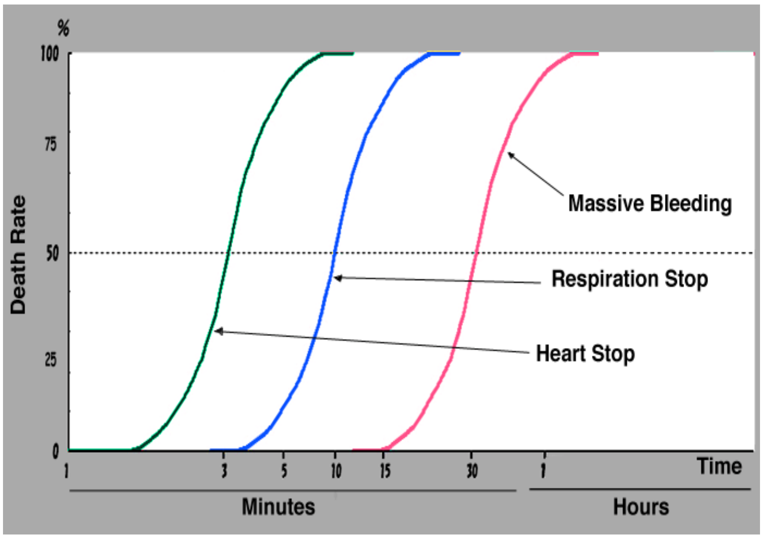


Figure 2.1: Golden Hour Principle
[1]

The golden hour is not simply constrained to traumatic emergency circumstances. This first hour of rising medical care is additionally extremely paramount in circumstances, for example, heart attack or stroke, where time is heart muscle or mind tissue. Developing medical intercessions can have a significant effect on a patient's survival and extreme capability to capacity. Patients and friends and family need to be mindful that not all healing facilities can offer definitive medical care. For patients who land at an emergency division that is unable to give the rising medical care that is required, the exchange methodology will start. The time that is squandered throughout the exchange procedure can prompt disaster.

But in India, we are still not able to provide medical assistance to all patients within very short time. In lack of knowledge people generally use first aid which locally known or web search for instant suggestion. The main limitation is that this information is generally from non-authentic sources.

Chapter 3

Information Retrieval and Natural Language Processing

This chapter introduce "information retrieval and natural language processing" to give some background on these two subjects which are central to this research.

3.1 Information Retrieval

In the standard IR model both the information need and the documents must first be translated into some alternate representation. Given figure shows the first step of the basic IR process where documents and information needs are translated into a new representation. In most cases, the information need is translated by a user into a query. Queries are often keyword queries sometimes with Boolean or phrase operators.

Some IR models extend the representation of the information need beyond the original keywords. This can be done by augmenting the original query terms with additional terms added found in a variety of ways such as query expansion, relevance feedback, or machine-readable thesauri. The representation used in this research retains the original query terms but augments them with linguistic tags. The text representation is augmented similarly.

This research differs from other approaches to IR in how queries are represent. The information need is represented as a natural language query. It is often a one-sentence description of the information need. Which will further converted into a Native Sub-Query, which will serves as an input for other Engines.

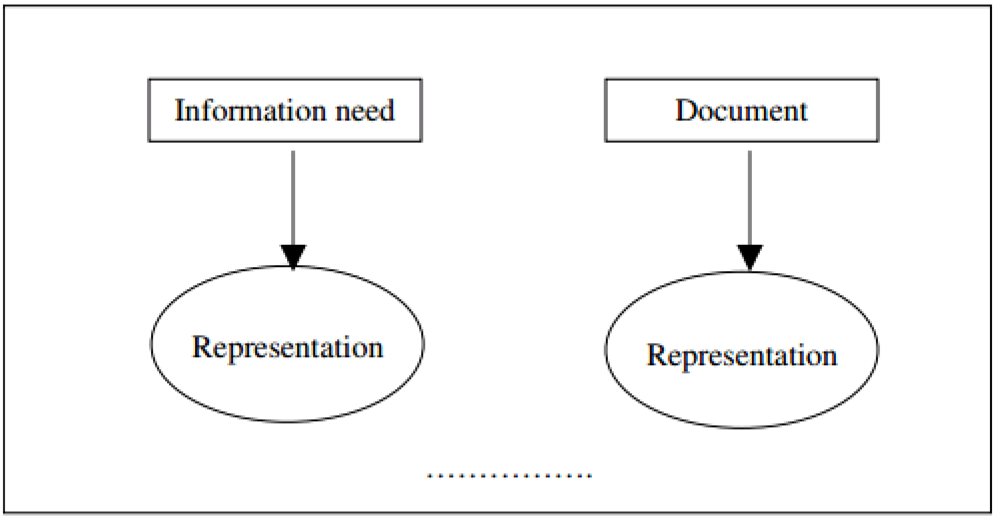


Figure 3.1: First step of Basic IR process
[2]

3.2 Natural Language Processing

Much of the work that has been done in "natural language processing (NLP)" has not been applied to IR. The extent that most search engines use NLP is to match morphological variants of query terms. In this section, we discuss some of the NLP techniques we apply to IR. These include syntactic parsing, 'part-of-speech (POS) tagging', 'named entity recognition', and others.

NLP algorithms automatically extract information contained implicitly in the text. Before discussing the application of these NLP techniques to information retrieval, it is important to understand the state-of-the-art NLP research and how much information we can hope to extract automatically from language in text format.

Much NLP research has been to identify syntactic features in language. Parsing, chunking, POS tagging, lexical cohesion are all examples of syntactic analysis. Since it seems that humans need background knowledge to comprehend the meaning of a sentence, it should not be surprising. Some researchers in the field see syntactic NLP as low hanging fruit which has been the primary focus of NLP researchers at the expense of research in semantic NLP. This section provides an overview of the NLP techniques that have been used in this research. These include only syntactic NLP techniques such as 'part-of-speech tagging', 'shallow parsing', 'stemming' and 'lexical cohesion' . Here semantic NLP techniques are not included because it is out of the scope of my research.

3.2.1 Syntactic NLP Techniques

Part-of-Speech Tagging:

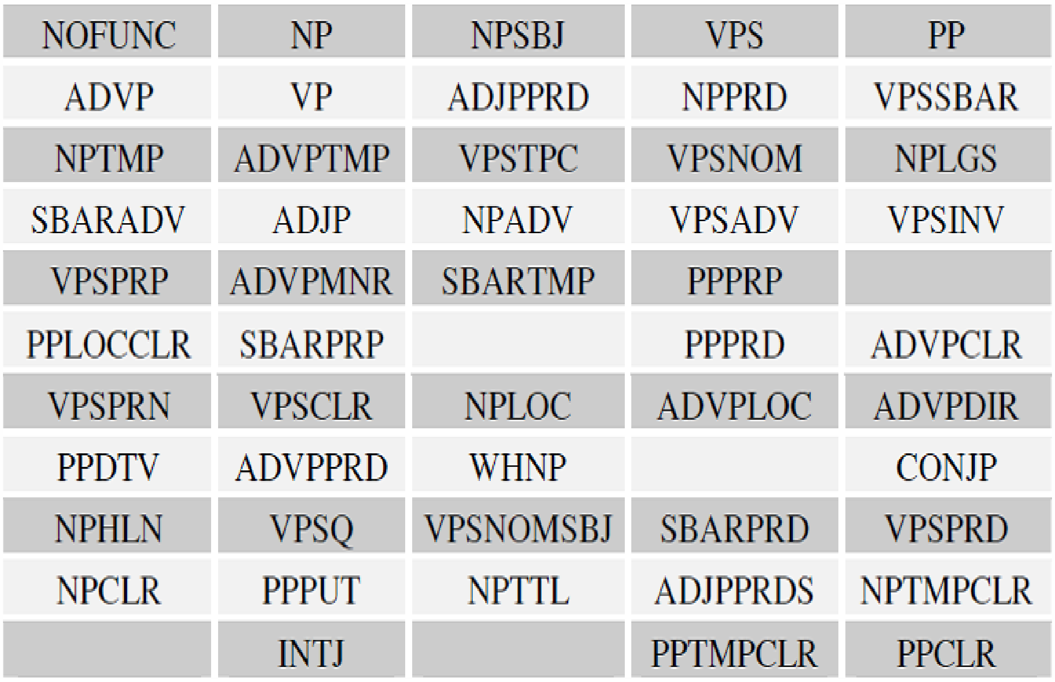
Part-of-speech may be one of the most basic forms of syntax. All languages have Parts-of-speech, although the word classes vary from language to language. The reader is likely familiar with broadest part-of-speech word classes such as verb, adjective, noun, etc. These word classes can be further broken down many times over (such as into plural nouns, infinitive verbs, etc.). In English, there is no canonical set of word classes. Linguistic topologists generally formulate tests to judge what part-of-speech a particular word is. However, even these linguistic tests can discover conflicting evidence for several parts-of-speech. For example, the English word many shows the behavior of determiner, predeterminer, and adjective.

Syntactic parsing:

Whereas parts-of-speech may be the most basic level of syntax, a full syntactic parse contains the most syntactic information. The high information content in a syntactic parse tree may be overwhelming for many computer applications, including information retrieval. In a full-parse, each word is a member of numerous constituent structures which are not readily collapsible into a single compact description of that constituent structure. Below Figure shows one way of describing syntax and grammatical roles in a compact form. These are syntactic parse tags used in PhraseNet.[9]

A word in a parse tree can be described by the depth, the words in the same constituent structure, the head word of that structure, the type of constituent structure, and other information about the structure that it modifies or attaches to. Thus, for information retrieval it is important that we choose only features that can be used effectively.

One way of using all the information in a parse tree for information retrieval would be to treat each constituent structure as a section of the document. This would allow the user to search for terms that occur in any specified type of constituent (e.g. nouns phrase or s-bar) and search for terms that co-occur in the same constituent structures. One could imagine a document scoring function that weights more highly query terms that co-occur in smaller constituent structures (this would be similar to a scoring function based on term proximity but would be proximity within the parse tree). For example, given the parse:



NOFUNC	NP	NPSBJ	VPS	PP
ADVP	VP	ADJPPRD	NPPRD	VPSSBAR
NPTMP	ADVPTMP	VPSTPC	VPSNOM	NPLGS
SBARADV	ADJP	NPADV	VPSADV	VPSINV
VPSPRP	ADVPMNR	SBARTMP	PPPRP	
PPLOCCLR	SBARPRP		PPPRD	ADVPCLR
VPSPRN	VPSCLR	NPLOC	ADVPLOC	ADVPDIR
PPDTV	ADVPPRD	WHNP		CONJP
NPHLN	VPSQ	VPSNOMSBJ	SBARPRD	VPSPRD
NPCLR	PPPUT	NPTTL	ADJPPRDS	NPTMPCLR
	INTJ		PPTMPCLR	PPCLR

Figure 3.2: Syntactic Parse Tags

((Jack and Jill)NP (went (up (the hill)NP)PP)VP)S.

We could treat each phrase as a section of the document. Picture the sentence as a structured document with section and subsection. Then section S is the entire document, the first NP is section 1, VP section 2, PP section 2.1 and so on. In this way, we can directly apply methods for structured retrieval to linguistic retrieval.

Shallow parsing:

We do not attempt to index an entire syntactic parse of each sentence as discussed in the previous section. Rather, as a simplification of the syntactic parse, we only consider the main constituent structures of a sentence as chunks of the sentence. This is in effect a shallow parse.[3] Thus, a sentence may be partitioned off into a 'noun phrase (NP)' subject, trailed by a 'verb phrase (VP)', and perhaps ending with a prepositional phrase (PP). This type of NLP may be particularly useful for IR since it breaks a sentence down into a few manageable chunks without greatly increasing the amount of data. One use of a shallow parse in IR would be to use a sentence scoring function to give a higher score to words that often use in input sentences.

Stemming and lemmatization:

For linguistic reasons, archives are going to utilize diverse types of an expression, for example, compose, composes, and sorting out. Furthermore, there are groups of derivationally related words with comparative implications, for example, majority rule government, fair, and democratization. As a rule, it appears to be as though it might be valuable for a quest for one of these words to return records that hold an alternate word in the set. Stemming typically alludes to an unrefined heuristic process that chops off the finishes of words in the trust of attaining this objective effectively more often than not, and frequently incorporates the evacuation of derivational appends. Lemmatization normally alludes to doing things legitimately with the utilization of a vocabulary and morphological analysis of words, typically expecting to evacuate inflectional endings just and to furnish a proportional payback or lexicon type of a saying, which is known as the lemma.

Lexical Cohesion:

Cohesive lexical units are multi-word terms that together function as a member of the lexicon. An example of this is fire hydrant in which the meaning is not obviously inferred from the words in the phrase alone. A certain amount of background knowledge is needed to understand the phrase. For many of these phrases it can be assumed that speakers of the language have a lexical entry for that term. In previous unpublished research this author showed that using a variety of information theoretical methods to identify cohesive multi-word lexical units. By automatically identifying "multi-word lexical units" in both the corpus and queries, we increase precision by ranking higher documents that contain the multi-word search terms as a unit rather than just matching documents that contain both words.

Chapter 4

Related Research

There are several search engine exist to process Natural Language and also some projects were carried out in this area. Here I will discuss some of them which I found related.

4.1 Stanford CoreNLP

Stanford Corenlp [4] gives a set of natural language dissection instruments which can take crude English language content enter and give the base manifestations of words, their parts of discourse, whether they are names of organizations, individuals, and so forth., standardize dates, times, and numeric amounts, and stamp up the structure of sentences as far as expressions and word conditions, and show which thing expressions allude to the same elements. Stanford Corenlp is an incorporated system, which make it simple to apply a pack of language investigation devices to a bit of content. Beginning from plain content, you can run all the instruments on it with only two lines of code. Its breaks down give the foundational building squares to larger amount and area particular content comprehension provisions.

Stanford Corenlp coordinates all our NLP devices, including the grammatical form (POS) tagger, the named entity recognizer (NER), the parser, and the coreference determination framework, and gives model records to examination of English. The objective of this undertaking is to empower individuals to rapidly and effortlessly get complete linguistic annotations of characteristic dialect writings. It is intended to be very adaptable and extensible. With a solitary alternative you can change which apparatuses ought to be empowered and which ought to be handicapped.

4.2 WordNet

Wordnet [5][6] is a vast lexical database of English. Things, verbs, modifiers and qualifiers are assembled into sets of cognitive equivalent words (synsets), each one communicating an unique idea. Synsets are interlinked by method for theoretical semantic and lexical relations. The ensuing system of genuinely related words and ideas might be explored with the program. Wordnet is likewise uninhibitedly and freely accessible for download. Wordnet's structure makes it a valuable device for computational linguistics and characteristic dialect handling.

Wordnet superficially resembles a thesaurus, in that it gatherings words together focused around their implications. Then again, there are some essential refinements. To start with, Wordnet interlinks word structures series of lettersas well as particular faculties of words. Subsequently, words that are found in close nearness to each other in the system are semantically disambiguated. Second, Wordnet names the semantic relations among words, inasmuch as the groupings of words in a thesaurus does not take after any unequivocal example other than significance comparabilit.

4.3 Natural Language Toolkit(NLTK)

NLTK [7] is a heading stage for building Python projects to work with human language information. It gives simple to-utilize interfaces to in excess of 50 corpora and lexical resources, for example, Wordnet, alongside a suite of content transforming libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

On account of an involved aide presenting programming basics nearby themes in computational linguistics, NLTK is suitable for linguists, engineers, learners, instructors, analysts, and industry clients apparently equivalent. NLTK is accessible for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, group driven task.

NLTK has been called ”a magnificent instrument for educating, and working in, computational linguistics utilizing Python,” and ”a stunning library to play with regular dialect.” Natural Language Processing with Python gives a pragmatic prologue to programming for dialect handling. Composed by the inventors of NLTK, it manages the onlooker through the basics of composing Python projects, working with corpora, ordering content, analyzing linguistic structure, and that's just the beginning.

4.4 PhraseNet

PhraseNet, [9] developed by Yuancheng Tu, Xin Li, and Dan Roth at University of Illinois, is an example of a context-sensitive lexical semantic knowledge system (Tu 2006). PhraseNet disambiguates WordNet word senses based on the context in which they are found. The context in this case could consist of both the syntactic structure of the sentence (e.g. Subject-Verb as in he ran or Subject- Verb-Object-PrepPhr as in he gave it to me) and words in the sentence.

This can be used to disambiguate the word sense for fork in they ate a cake with a fork since in that context the object of the preposition is usually either a utensil (e.g. spoon or fork) or a food (as in cake with strawberries). Thus, we know that the word sense for fork is utensil not as in fork in the road. In this example, disambiguating the word sense also tells you what the PP attaches to (if it is a utensil it attaches to the verb, if it is food is attaches to the object cake). PhraseNet is different from previous approaches in its novel use of WordNet to do word sense disambiguation.

4.5 OpenNLP

The Apache Opennlp [10] library is a machine taking in based Java Toolkit for the transforming of natural language content. It underpins the most widely recognized NLP tasks, for example, tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. These tasks are typically needed to fabricate more praiseworthy content transforming administrations. Opennlp likewise incorporates maximum entropy and perceptron based machine learning.

4.6 Gensim

Gensim [11] is an open-source vector space modeling and theme modeling tool compartment, actualized in the Python programming dialect, utilizing quick C and FORTRAN framework libraries for execution. It is particularly expected for taking care of extensive content accumulations, utilizing productive online calculations. Gensim incorporates executions of tfidf, random projections, deep learning with Google's word2vec algorithm (reimplemented and optimized in Cython), latent semantic analysis (LSA) and latent dirichlet allocation (LDA), including circulated parallel variants.

4.7 General Architecture for Text Engineering (GATE)

General Architecture for Text Engineering or GATE is a Java suite of instruments originally created at the University of Sheffield starting in 1995 and now utilized worldwide by a wide group of researchers, organizations, instructors and learners for different kinds of natural language processing assignments, incorporating information extraction in numerous languages.

GATE incorporates an information extraction framework called ANNIE (A Nearly-New Information Extraction System) which is a situated of modules including a tokenizer, a gazetteer, a sentence splitter, a grammatical form tagger, a named substances transducer and a coreference tagger. ANNIE might be utilized as-is to give fundamental information extraction usefulness, or give a beginning stage to more particular errands. Languages right now took care of in GATE incorporate English, Spanish, Chinese, Arabic, Bulgarian, French, German, Hindi,italian, Cebuano, Romanian, Russian.

Chapter 5

Language Modeling

This chapter gives information about various models which will be useful for assigning labels on the bases of features involved. At the point when utilized as a part of information retrieval, a language model is connected with a report in a collection. With query Q as input, recovered records are positioned focused around the probability that the archive's language model might produce the terms of the query. Here this knowledge will help to categories diseases on the basis of symptoms.

5.1 Naive Bayes classifiers

In Naive Bayes classifiers [12], each characteristic gets a say in figuring out which label ought to be allocated to a given information esteem. To pick a label for an information esteem, the credulous Bayes classifier starts by figuring the earlier probability [13] of each one label, which is dictated by checking frequency of each one label in the preparation set. The commitment from each one characteristic is then consolidated with this earlier probability, to touch base at a probability gauge for each one label. The label whose likelihood estimate is the highest is then assigned to the input value. Following figure illustrates this process.

Figure 5.1 is an unique representation of the technique utilized by the Naive Bayes classifier to pick the category for data inquiry. At first our classifier begins at a point closer to the "Mellow" label. Anyway it then recognizes the impact of each one characteristic. In this illustration, the information inquiry holds the saying "endure" which is a frail marker for Moderate, yet it likewise holds the expression "heart attack" which is a solid

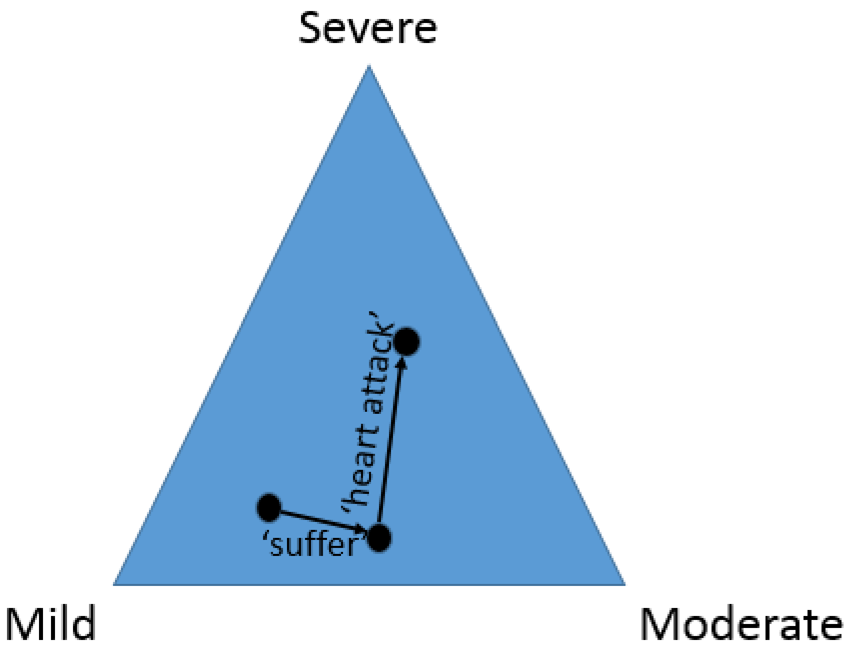


Figure 5.1: An illustration of Naive Bayes procedure

pointer for Severe conditions. After every characteristic has made its commitment, the classifier checks which label it is closest to, and doles out that label to the information.

To produce a labeled input, the model first picks a label for the input, then it creates each of the input's features focused around that label. Each one feature is thought to be totally independent of one another feature, given the label. In view of this presumption, we can compute a statement for P(label/features), the likelihood that an input will have a specific label given that it has a specific set of features. To pick a label for an alternate input, we can then basically pick the label l that amplifies P(l/features). This might be figured by the given declaration:

$$P(label|features) = P(features, label)/P(features)$$

Which we'll call the label likelihood.

5.2 Maximum Entropy Classifiers

The Maximum Entropy classifier [14] utilizes a model that is very much alike to the model utilized by the naive Bayes classifier. But instead than utilizing probabilities to set the model's parameters, it uses seek systems to discover a set of parameters that will expand the execution of the classifier. Specifically, it searches for the set of parameters that maximizes the aggregate probability of the training corpus, which is characterized as:

$$P(features) = \sum x|in|corpusP(label(x)|features(x))$$

Where P(label—features), the probability that an input whose features are features will have class label label, is defined as:

$$P(label|features) = P(label, features)/\sum labelP(label, features)$$

Due to the conceivably perplexing connections between the impacts of related characteristics, there is no real way to straightforwardly figure the model parameters that augment the probability of the preparation set. Therefore, Maximum Entropy classifiers pick the model parameters utilizing iterative optimization techniques, which introduce the model's parameters to irregular values, and afterward over and again refine those parameters to bring them closer to the ideal result. These iterative optimization techniques ensure that every refinement of the parameters will bring them closer to the ideal qualities, however don't essentially give a method for deciding when those ideal qualities have been arrived at. Since the parameters for Maximum Entropy classifiers are chosen utilizing iterative optimization techniques, they can take quite a while to learn. This is particularly genuine when the measure of the preparation set, the amount of characteristics, and the amount of names are all expansive.

5.3 Generative vs Conditional Classifiers

A critical distinction between the naive Bayes classifier and the Maximum Entropy classifier concerns the sort of inquiries they might be utilized to reply. The naive Bayes classifier is a case of a generative classifier, which assembles a model that predicts P(input, label), the joint probability of an (input, label) pair. Accordingly, generative models might be utilized to answer the accompanying inquiries:

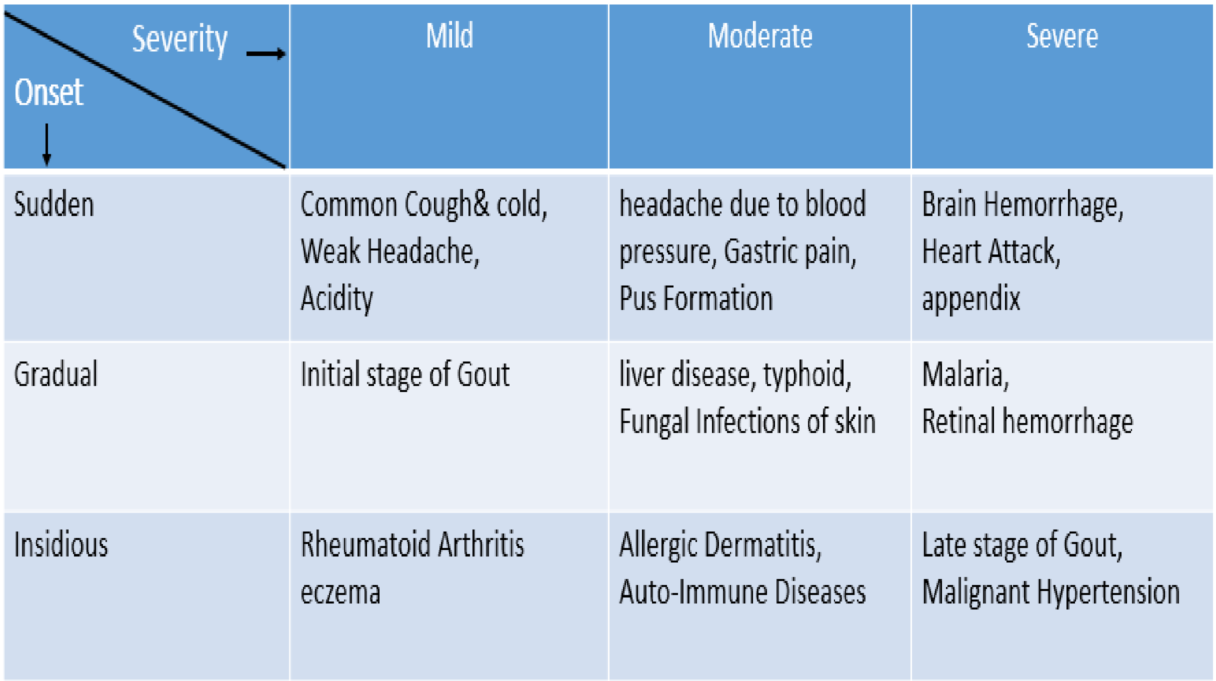
1. What is the most likely label for a given input?

2. How likely is a given label for a given input?
3. What is the most likely input value?
4. How likely is a given input value?
5. How likely is a given input value with a given label?
6. What is the most likely label for an input that might have one of two values (but we don't know which)?

The Maximum Entropy classifier, then again, is a case of restrictive classifier. Restrictive classifiers construct shows that foresee P(label/input) the likelihood of a label given the input esteem. Along these line, restrictive models can in any case be utilized to answer addresses 1 and 2. In any case, contingent models can't be utilized to answer the remaining inquiries 3-6.

All in all, generative models are strictly more capable than conditional models, since we can ascertain the conditional likelihood P(label/input) from the joint likelihood P(input, label), however not the other way around. Nonetheless, this extra power includes some significant downfalls. Since the model is all the more compelling, it has all the more "free parameters" which need to be taken in. In any case, the extent of the preparation set is settled. Therefore, when utilizing an all the more influential model, we wind up with less information that might be utilized to prepare every parameter's quality, making it harder to discover the best parameter values. Subsequently, a generative model may not benefit as a vocation at noting inquiries 1 and 2 as a conditional model, since the conditional model can center its deliberations on those two inquiries. In any case, in the event that we do need replies to inquiries like 3-6, then we have no decision yet to utilize a generative model.

The refinement between a generative model and a conditional model is undifferentiated from the differentiation between a topographical map and a picture of a horizon. Despite the fact that the topographical map could be used to answer a more far reaching mixed bag of queries, it is essentially more troublesome to produce an exact topographical map than it is to create a precise horizon.



Severity → Onset ↓	Mild	Moderate	Severe
Sudden	Common Cough& cold, Weak Headache, Acidity	headache due to blood pressure, Gastric pain, Pus Formation	Brain Hemorrhage, Heart Attack, appendix
Gradual	Initial stage of Gout	liver disease, typhoid, Fungal Infections of skin	Malaria, Retinal hemorrhage
Insidious	Rheumatoid Arthritis eczema	Allergic Dermatitis, Auto-Immune Diseases	Late stage of Gout, Malignant Hypertension

Figure 5.2: Disease Category

Chapter 6

The Proposed Algorithm

The prime purpose of this project is to generate Native Sub-Query from user query. To achieve this goal inputted user query passes through various phases. In the initial phase Syntactic NLP techniques are applied on the user query and at later phase slightly modified version of Naive Bayes classification algorithm is applied on it. System also performs some network processing activity to gather some other important parameters.

6.1 Phase I: Appling NLP Techniques with Network Processing

- Wrote python scripts for basic NLP tasks like Tokenization, Lexical Cohesion and Stemming etc.

- Use some convenient algorithms for tasks like Part of Speech tagging and spell checking of tokens.

- Removed all the unnecessary Tokens like stop words and keep only those which will be used later in classification because those words/tokens are insignificant for classification.

- Extract location parameters like City, Latitude and Longitude of user with some other parameters like Date and Time at which query fired by the user, using their connection information.

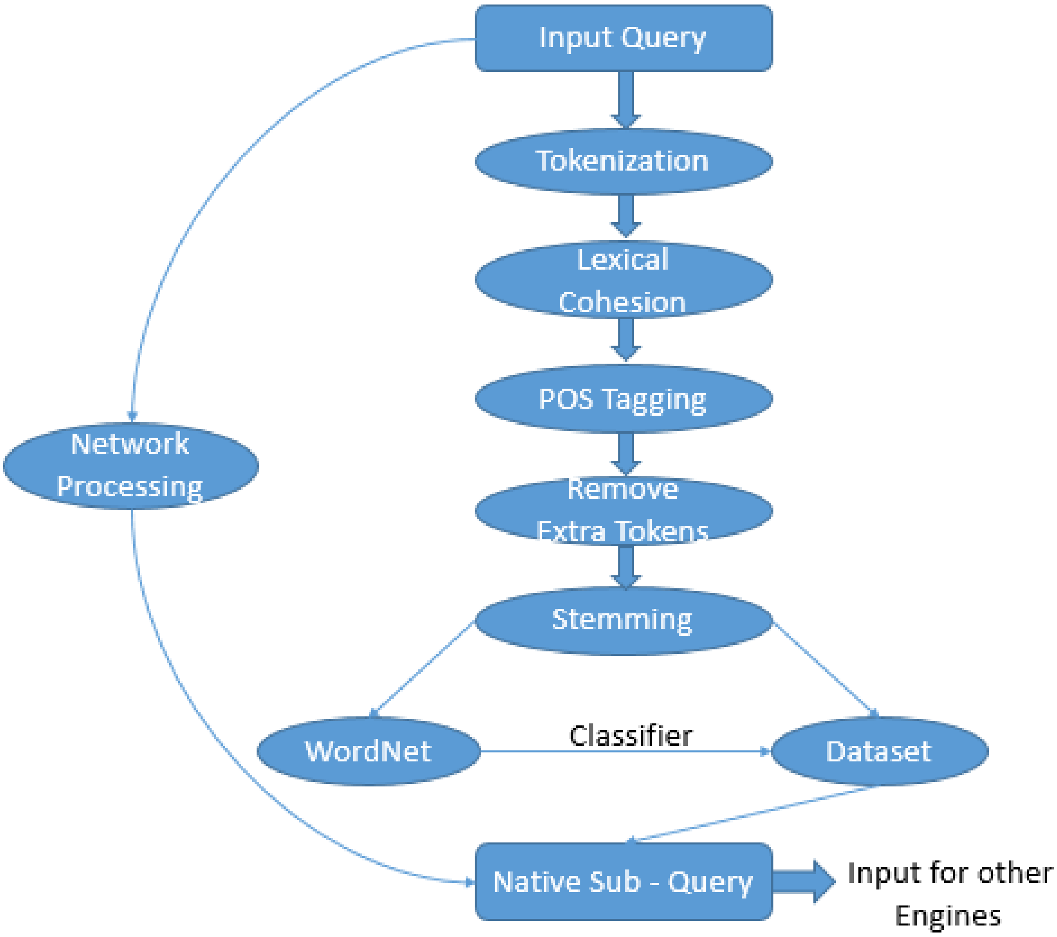


Figure 6.1: Proposed Algorithm

6.2 Phase II: Classification Algorithm for Naive Bayesian method

A slightly modified version of Naive Bayes classification algorithm is applied on the remained terms obtained as output of phase-I. This algorithm only assigns label for appropriate disease category from categories Mild, Moderate and Severe to user query with highest probability difference P(Category/ Q) between all three categories, where Q is unlabeled user query. Same algorithm will apply for gender (Male/Female) determination.

For a given new query, for predicting category label Nave Bayes method follows this algorithm:

Step1: Calculation of P(Query/Category):

1. P(Query/Category) = 1

2. for each term qi belongs to query:

3. P(Query/Category) = #no of times t occurred in category / #total no of occurrences of category

Step2: Calculation of P(Category)

P(Category) = #no of terms in category / #no of terms all categories

Step3: Finding P(Category/Query)

Since P(Document) is same for all categories we can ignore that.

P(Category/Query) = P(Query/Category) * P(Category)

Step4: Predicting new category for new unseen query

Category of new query = argmax P(Category/Query)

This algorithm also provide 'Self Learning feature', means: for a given query, if this classifier predicts any label among Mild, Moderate and Severe with high probability, then this prediction will be used as a training example and will appended in training data.

6.3 How It Works

- Each and every feature makes its contribution in deciding categories by "voting against" labels which generally doesn't occurs with that category.

- The likelihood score will be reduced by the factor of multiplication with the probability that the given input will has that feature with this particular label.

- For example:

-"if the word 'Weakness' occurs in 12% of the 'Mild' documents, 10% of the 'Moderate' documents and 2% of the 'Severe' documents, then the likelihood score for the Mild label will be multiplied by 0.12; the likelihood score for the Moderate label will be multiplied by 0.1, and the likelihood score for the Severe label will be multiplied by 0.02. The overall effect will be to reduce the score of the 'Severe' label significantly more than the other two categories."

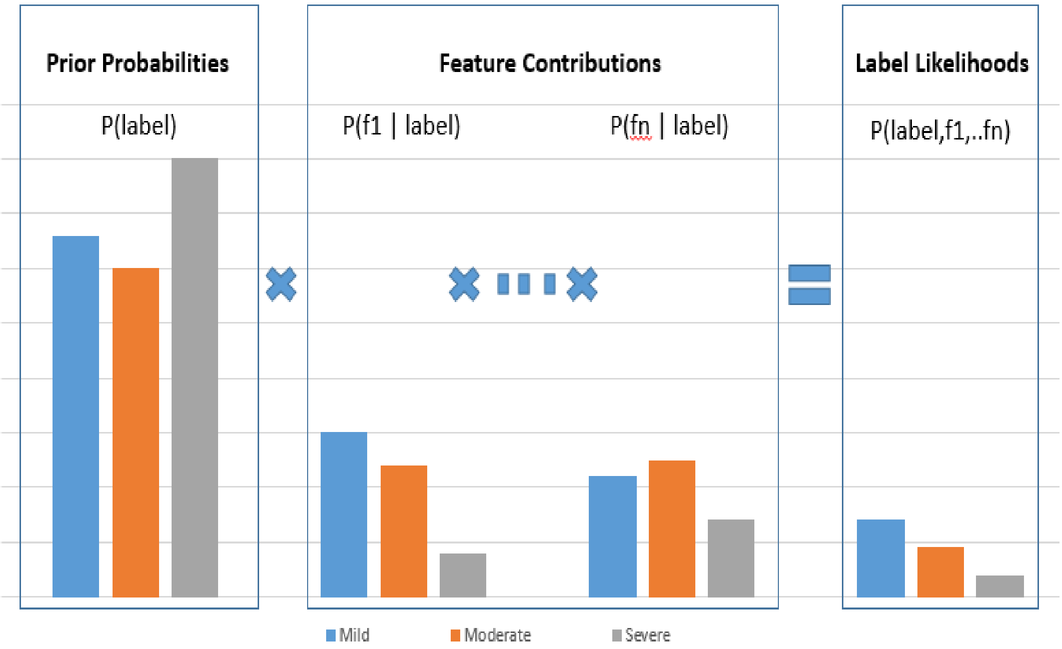


Figure 6.2: Working of NB Algorithm

Chapter 7

Implementation and Results

This chapter covers the details of the implantation approach followed to generate Native Sub-Query. Here we take an example query and process it through various phases, which are already discussed in previous chapters. The primary implementation of the project is performed using Python 2.7 and NLTK 2.0.

User Query:

"My brother is suffering from heart attack"

Phase-I:

1. Tokenization:

   Input = 'My brother is suffering from heart attack'

   Output = ['My', 'brother', 'is', 'suffering', 'from', 'heart', 'attack']

2. Lexical Cohesion:

   Input = ['My', 'brother', 'is', 'suffering', 'from', 'heart', 'attack']

   Output = ['My', 'brother', 'is', 'suffering', 'from', 'heart attack']

3. POS Tagging:

Input = [’My’, ’brother’, ’is’, ’suffering’, ’from’, ’heart attack’]

Output = [(’My’, ’PRP$’), (’brother’, ’NN’), (’is’, ’VBZ’), (’suffering’, ’VBG’), (’from’, ’IN’), (’heart attack’, JJ’)]

4. Remove Extra Tokens:

Input = [(’My’, ’PRP$’), (’brother’, ’NN’), (’is’, ’VBZ’), (’suffering’, ’VBG’), (’from’, ’IN’), (’heart attack’, JJ’)]

Output = [brother, ’suffering’, ’heart attack’]

5. Stemming:

Input = [brother, ’suffering’, ’heart attack’]

Output = [brother, ’suffer, ’heart attack’]

After completion of the above steps we left with few terms which will help us in deciding disease category and retrieving other important information. Now these values(Tokens) passes to WordNet for gathering relative information.

6. WordNet:

’brother’ :

[Synset(’brother.n.01’), Synset(’brother.n.02’), Synset(’buddy.n.01’), Synset(’brother.n.04’), Synset(’brother.n.05’)]

brother.n.01: a male with the same parents as someone else brother.n.02: a male person who is a fellow member (of a fraternity or religion or other group)

buddy.n.01: a close friend who accompanies his buddies in their activities

brother.n.04: used as a term of address for those male persons engaged in the same movement

brother.n.05: (Roman Catholic Church) a title given to a monk and used as form of address

'suffer':

[Synset('suffer.v.01'), Synset('suffer.v.02'), Synset('suffer.v.03'), Synset('digest.v.03'), Synset('suffer.v.05'), Synset('suffer.v.06'), Synset('hurt.v.06'), Synset('suffer.v.08'), Synset('suffer.v.09'), Synset('suffer.v.10'), Synset('suffer.v.11')]

suffer.v.01: undergo or be subjected to

suffer.v.02: undergo (as of injuries and illnesses)

suffer.v.03: experience (emotional) pain

digest.v.03: put up with something or somebody unpleasant

suffer.v.05: get worse

suffer.v.06: feel pain or be in pain

hurt.v.06: feel physical pain

suffer.v.08: feel unwell or uncomfortable

suffer.v.09: be given to

suffer.v.10: undergo or suffer

suffer.v.11: be set at a disadvantage

'heart attack':

S: (n) heart attack (a sudden severe instance of abnormal heart function)

This information will further processed in phase-II to obtain the required Native Sub-Query.

Phase II:

- Train-set = training set consist of medical documents related to all three categories, stored in three different files.
- Features-set = features are extracted from training data which also include some external keywords.
- Test-set = tested on Sixteen Hundred Seventy Nine processed user queries. Eg. [‘brother’, ‘suffer’, ‘heart attack’]”
- Accuracy Achieved = 87.31

Generated Native Sub Query:

nquery :

[

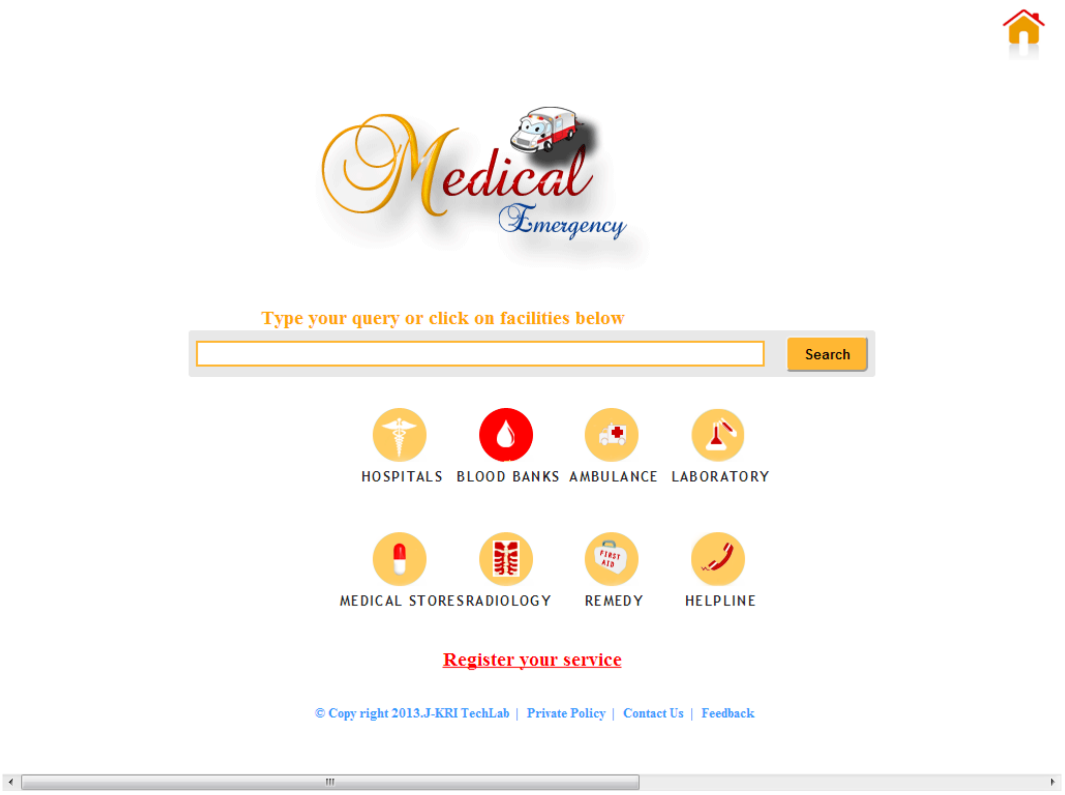
{’City’: ’Pune, Maharashtra, India’, ’Latitude’: 18.52912, ’Longitude’: 73.8737, ’Time’: ’3 PM’, ’Date’: ’11-04-2014’},

{’Category’: ’Severe, ’Gender’: ’Male’, ’Age’: ’NA’, ’Disease’: ’heart attack’, ’Query’: My brother is suffering from heart attack ’}

]

–Probable disease recognition module is not yet implemented.

User Interfaces:

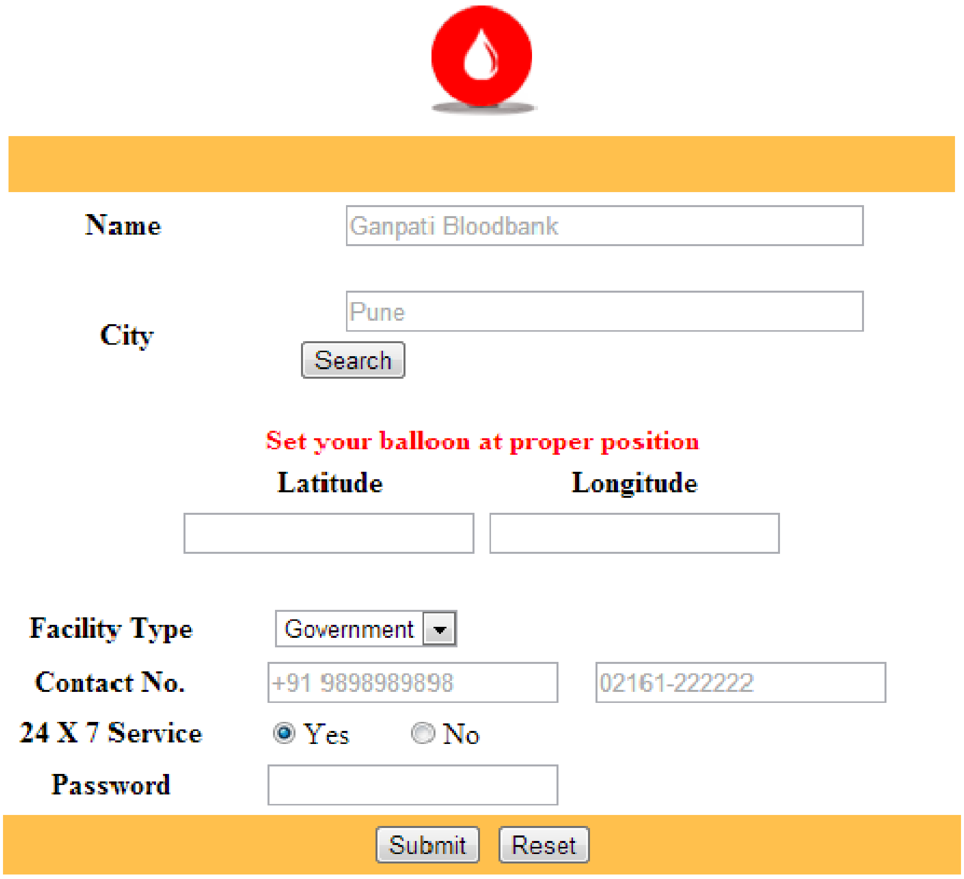


Type your query or click on facilities below

Register your service

© Copy right 2013.J-KRI TechLab | Private Policy | Contact Us | Feedback

Figure 7.1: Home Page



Name Ganpati Bloodbank

City Pune

Set your balloon at proper position

Latitude **Longitude**

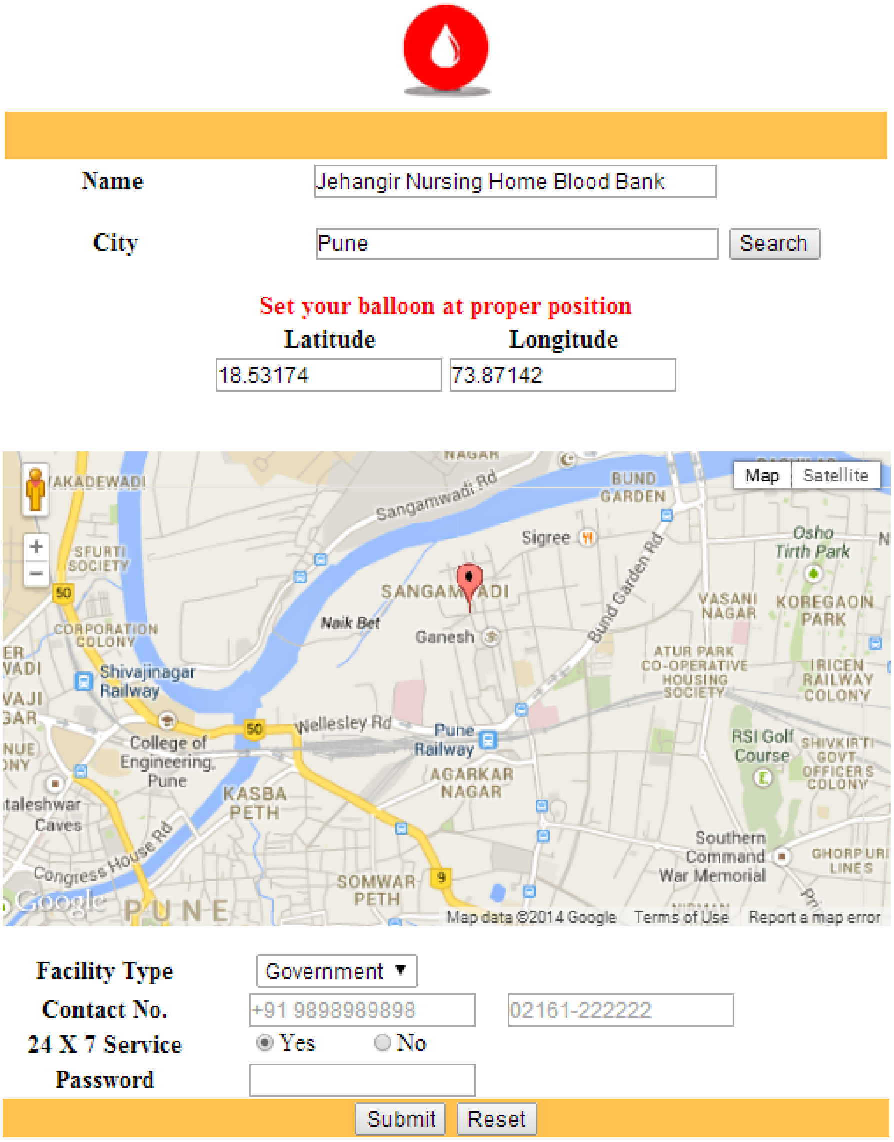
Facility Type Government

Contact No. +91 9898989898 02161-222222

24 X 7 Service ● Yes ○ No

Password

Figure 7.2: Facility Registration: (Blank Form)



Name	Jehangir Nursing Home Blood Bank
City	Pune [Search]

Set your balloon at proper position

Latitude	Longitude
18.53174	73.87142

Facility Type	Government ▼	
Contact No.	+91 9898989898	02161-222222
24 X 7 Service	● Yes ○ No	
Password		

[Submit] [Reset]

Figure 7.3: Facility Registration: (Filled Form)

Chapter 8

Conclusion and Future Work

This thesis uses the basic concepts of information retrieval and applies them on single line natural language query. In this thesis one liner queries are processed instead of large set of documents and it tries to extract as much as possible information from a sentence submitted by user as a query. This information will play a vital role in assisting users/patients in situations of medical emergency for making evidence based medical decision within time constraints.

Future Work :

- Optimizing system performance by using the additional information gathered through WorldNet.
- Developing some effective mechanism for merging qualified user queries with training data.

Bibliography

[1] http://www.medindia.net/patients/patientinfo/traumagoldenhour.htm

[2] B. E. Lambert, ”Improving information retrieval with natural language processing”, University of Massachusetts Amherst, USA, 2003.

[3] Siasar djahantighi F. , Norouzifard M., Davarpanah S.H. and Shenassa M.H. , Using Natural Language Processing in Order to Create SQL Queries, In Proceedings of the International Conference on Computer and Communication Engineering 2008, Kuala Lumpur, Malaysia.

[4] http://nlp.stanford.edu/

[5] George A. Miller, ”WordNet: a lexical database for English, Princeton University”, Princeton, New Jersey, USA, 1995.

[6] Princeton University ”About WordNet.” WordNet. Princeton University. 2010. ¡http://wordnet.princeton.edu¿

[7] http://nltk.org/

[8] D. Vadas, and J. R. Curran, ”Programming with Unrestricted Natural Language”, School of Information Technologies, University of Sydney NSW, Australia, 2006.

[9] Tu, Yuancheng, Xin Li, Dan Roth, PhraseNet: towards context sensitive lexical semantics., Proc. of the Annual Conference on Computational Natural Language Learning, 2003.

[10] http://opennlp.apache.org/

[11] http://radimrehurek.com/gensim/

[12] Langley Pat, Iba Wayne and Thomas K. 1992. An analysis of Bayesian classiers. In Proceedings of the Tenth National Conference of Articial Intelligence. AAAI Press. 223-228.

[13] Bird Steven, Klein Ewan and Loper Edward, Natural Language Processing with Python – Analyzing Text with the Natural Language Toolkit, O'Reilly Media, 2009.

[14] Adam L. Berger , Vincent J. Della Pietra , Stephen A. Della Pietra, A maximum entropy approach to natural language processing, Computational Linguistics, v.22 n.1, p.39-71, March 1996.