CHAPTER -5
NAMED ENTITY RECOGNITION

5.1 INTRODUCTION

Named Entity Recognition is a sub task of Information Extraction. It is a necessary process in various natural language processing applications. Recent Research community gives more concentration on the Named Entity Recognition[13]. The progress of Named Entity Recognition reached a new height in various domains such as newswires, biomedical etc..

Entity extraction and entity identification are also known as Named Entity Recognition. The process of Named Entity Recognition is identifying and categorizing the proper names and phrases available in the given text.

There is a rapid increase in the use of Ontology in the field Computer Science and Information Technology in recent research works, especially in the concept of semantic web. The ontology’s are strongly associated with semantic web[29]. From the large corpora of documents, ontology elements are extracted with the help ontology learning, which is a automatic process. Named Entity Recognition is used to generate automatically an ontology from a large collection of documents.
The Named Entity Recognition system deals with various issues such as:

i. Identification of Named entity Recognition from the text.

ii. Categorizing the named entities to the appropriate classes.

iii. If more than one synonym is there for identified entities then suitable name of the identified entity has to be found.

The task of Named Entity Recognition is difficult in the early days due to:

i. The availability of millions of named entities.

ii. Continuously increasing numbers of entity names rapidly.

iii. Due to unavailability of naming agreement, an excessive use of short forms, the use of various synonyms, homonyms, and some biological names are multifaceted names that consist of numerous words

The techniques for identifying the named entities have been categorized into three different classes such as:

i. Dictionary-based techniques

ii. Rule-based techniques

iii. Machine learning techniques

5.1.1 DICTIONARY-BASED TECHNIQUES

The dictionaries will have huge collection of names of the entities. The phrases to be matched from the given text will be compared with
the existing entries in the dictionaries. The dictionaries are not be updated with the same speed with which the biomedical named entities are increasing on par with the result of research findings[33]. This leads to the difficulty in identifying the new and unknown entities from the given text. This method gives good rate of precision but low recall.

5.1.2 RULE-BASED TECHNIQUES

Hand crafted rules by human experts, lexical and linguistic rules will be used in the Rule-based techniques to find the named entities. The characteristics and the context of the named entities will be described using the rules. Manual way of writing the rules is a time consuming and laborious one[43]. The hand crafted rules will be specific to certain scenarios so it can be used only to identify known entities. Unknown entities apart from specifications cannot be identified. This kind of techniques generate high precision and low recall.

5.1.3 MACHINE LEARNING TECHNIQUES

In Machine learning, Named Entity Recognition technique is a kind of artificial intelligence method. This kind of Named Entity Recognition technique identifies the entities by searching for the patterns in the given text using algorithms [17]. The machine learning algorithms are classified as supervised and unsupervised. Supervised algorithms can relate what has been cultured in the precedent to new
data. Unsupervised algorithms can describe inference from the given dataset. The supervised machine learning system will have the training phase and testing phase. From the training data the supervised system will build a machine learning model [72][35]. Such properties are called features. These features are used to classify the classes of identified entities. The machine learning based Named Entity Recognition technique can perform better than the other Named Entity Recognition techniques.

5.2 ARCHITECTURE OF NAMED ENTITY RECOGNITION SYSTEM

The Named Entity Recognition system is a collection of various subtasks. The Fig: 5.2 elaborate the overall architecture of Named Entity Recognition system. The subtasks of Named Entity Recognition are:

- Preprocessing
- Syntactic Annotation
- Semantic Annotations
  - L-NER (Lexical Named Entity Recognition)
  - M-NER (Machine Learning Named Entity Recognition)

5.2.1 CORPUS OF RETRIEVED DOCUMENTS

The Named Entity Recognition is a subtask of information extraction. This process has a preceding process known as information retrieval process. In the information retrieval, from the corpus of
biomedical databases the documents have been retrieved using the IR model based on the user requirements[57]. The users specify their requirements in terms of keywords in the user interface design of the user requirement collection process. Depends on the user requirements the IR system will generate a Boolean query. The query will be given to the Enterz programming Utilities[11]. Based on the preference given by the user in the user requirement, the “Enterz Utility System” will retrieve the biomedical documents. The retrieved documents will be restructured into the text format using Apache PDFBox library which can be understood by the natural Language processing tools. The restructured documents will then be stored in the database. The relevance of the retrieved documents is validated using the Document – Weight Matrix.

5.2.2 PREPROCESSING

The preprocessing is the initial step in the process of Named Entity Recognition. The document will be separated in to basic text elements for succeeding processing by the text preprocessing module. Most of the text processing systems work on the sentences rather than on words or paragraphs[63]. So the preprocessing module splits the given abstract or whole documents in to single sentences. The preprocessing module consist of three sub modules such as:
- Sentence Splitting
- Tokenization
- Stopword Removal

The Tokenization process identifies various tokens of the given document. Sentence splitting process splits the documents into simple sentences using various criteria[37]. Stopword Removal removes the words that are not required.

Further in the process the tokens will be forwarded to the Named Entity recognition process for identification. Text preprocessing includes converting entity names using predefined rule set.

![Diagram of preprocessing steps]

**Fig: 5.2 Preprocessing of Named Entity Recognition**

**5.2.2.1 TOKENIZATION**

Tokenization is a prerequisite for any Named Entity Recognition Processes. The process of converting the input sentences as sequence of
tokens is known as tokenization. Tokenizing using white space and punctuation symbols is the simplest form of tokenization[19][41]. This simple method of tokenization will not work on Abbreviations, Notations and Equations. Also hyphenation creates the ambiguity to determine the tokens.

In biomedical tokenization of text simply by using white spaces would be neither sufficient nor suitable, due to (i) the diversity of the symbols and notations used in the domain, (ii) inconsistent usage of white spaces, (iii) occurrence of punctuation marks inside the Named Entities, (iv) inclusion of new terms, (v) use of scientific terminologies, (vi) unusual orthography, and (vii) continuation of indefinite punctuations.

For every rule, there will be a left hand side (LHS) part and a right hand side (RHS) part.

- The LHS is a regular expression which has to be matched on the input.
- The RHS describes the annotations to be added to the Annotation Set.
- The LHS is separated from the RHS by ‘>’.
- The following operators can be used on the LHS:

  | (or)

  * (0 or more occurrences)
? (0 or 1 occurrences)
+ (1 or more occurrences)

- The RHS uses ‘;’ as a separator and has the following format:
  \{LHS\} > \{Annotation type\};\{attribute1\}={value1};...;\{attribute n\}={value n}\}

The tokeniser rule for a word beginning with a single capital letter is:
‘UPPERCASE_LETTER’ ‘LOWERCASE_LETTER’* >Token;orth=upperInitial;kind=word;

It states that the string must begin with an uppercase letter, followed by zero or more lowercase letters. This sequence will then be annotated as type ‘Token’. The attribute ‘orth’ (orthography) has the value ‘upperInitial’; the attribute ‘kind’ has the value ‘word’.

**TYPES OF TOKEN**

**Word Token**

A word is known as any set of adjacent upper or lowercase letters, together with a hyphen. A word also has the attribute ‘orth’, for which the following values are defined:

- upperInitial - initial letter is uppercase, rest are lowercase
- allCaps - all uppercase letters
- lowerCase - all lowercase letters
- mixedCaps - any combination of upper and lowercase letters not included in the above categories.
Number

A number is defined as any grouping of consecutive digits. There are no subdivisions of numbers.

Symbol

Two types of symbol are defined: currency symbol (e.g. ‘$', '£') and other symbol (e.g. '&', ').

Punctuation

Three types of punctuation marks are defined:

- start_punctuation { e.g. ‘(’ }
- end_punctuation { e.g. ‘)’ }
- other punctuation { e.g. ‘:’ }

Every punctuation mark is a separate token.

Space Token

White spaces are divided into two types of Space Token viz., space and control - according to whether they are pure space characters or control characters. Any homogeneous set of space or control characters is defined as a Space Token.

5.2.2.2 SENTENCE SPLITTING

The process of determining the sentence boundaries is known as sentence splitting. It splits the given documents into sentences. The sentence splitting process is a prerequisite for various Natural Language Processing Processes. Identifying sentences with the help of
delimiters is a tough task due to the irregularity of the biomedical terms. Every splitted sentence will have an annotation tag known as 'sentence'. Each sentence splitter is a flow of finite-state transducers[21][26]. It splits the text into sentences. Sentence splitter is a prerequisite for the Parts Of Speech Tagger. The Regular Expression based sentence splitter works well for irregular kind of sentences. The system addresses three different types of encode patterns:

- **Internal Splits**
  Internal splits, splits the sentence based on the punctuation marks. It may be a part of a sentence.

- **External Splits**
  External splits, splits the sentence which is not part of any sentences. but successive lines.

- **Non Splits**
  Non splits text portions that might be seen as splits but they should be ignored. That is some biomedical terms and abbreviations may have punctuation mark in between as a part of a name.

### 5.2.2.3 STOPWORD REMOVAL

In text processing, Stopwords are words which are filtered away prior or after processing the text[55]. The set of commonly used words in English not belonging to entities are known as stop words. The cause
why stop words are serious to various applications is that, if we remove the words that are very usually used in a given language. A set of common English words are denoted as Stop words [67]. Stop words refer to the most common words in a language, which are filtered out before processing of natural language data. Sentences will be filtered[7]. Any matching tokens from the set will be filtered as Stopword’s.

**Types of Stop Words**

Stop words are generally thought to be a “single set of words”. Though stop words are common words, there will be no particular global list of stop words[16]. It really can mean different things to different applications. For example, in some applications removing all stop words right from determiners (e.g. the, a, an) to prepositions (e.g. above, across, before) to some adjectives (e.g. good, nice) can be an appropriate stop word list.

**Determiners** - Determiners likely to spot nouns where a determiner usually will be followed by a noun examples: the, a, an, another

**Coordinating conjunctions** - Coordinating conjunctions which attaches words, phrases, and clauses examples: for, an, nor, but, or, yet, so
**Prepositions** - Prepositions states temporal or spatial relations. Examples: in, under, towards, before. The most common list of stop words are enclosed in the appendix – I.

**5.2.3 Syntactic Annotation**

Syntactic annotation is including syntactic information to a corpus by integrating the information into the text pointer of syntactic constructions[33]. The method of evaluating a sequence of symbols is known as Syntactic analysis. It is a typical form of morphological annotation such as part-of-speech tagging, chunking which designates the grammatical category of every phrase of token in the corpus. Syntax is a combination of words, tokens and morphemes[73]. The various types of syntactic form are clauses, phrases etc., Morphology analysis is a study of how words are formed from smaller parts known as morphemes[45]. The syntactic form considers about phase structures and dependency grammar.

![Fig: 5.2.3 Syntactic Annotation](image)

- **POS Tagging**
- **Chunking**
The primary way of annotating the given text is grammatical tagging. Syntactic annotation is a prerequisite for semantic annotation. The process of syntactic annotation is more complex than semantic annotation. Part of speech tagging and Noun phrase chunking are the two subtasks of Syntactic annotation.

5.2.3.1 PARTS OF SPEECH TAGGING

The Parts-of-Speech tagging is the processes of categorizing the given text into appropriate parts-of-speech. In the process of parts-of-speech tagging the tokens are sequentially labeled using the syntactic labels. Tokenizer and Sentence splitter are the prerequisites for parts-of-speech tagging. The categorized feature taggers wear included to each tokens. The parts-of-speech tagging algorithms are classified into two different categories such as:

- Rule based part-of-speech tagging
- Stochastic part-of-speech tagging

The system includes Rule based Brill tagger for part-of-speech tagger, which is created by Eric Brill. Using the predefined rule set the tokens wear allocated with the appropriate tags.

The tokenization is a preprocessing task of Parts-of-Speech tagging. Initially the given text is processed, words into tokens. Initially the tokens wear tagged with primary tags using morphological rules. The Parts-of-Speech tagging is a complex process. The tokens will have
ambiguous and more than one meaning. The meaning of the tokens depend on the context of the text. The preliminary tagging process uses basic rules such as capitalization, prefix and suffix strings. In the next stage, after the applying the relative rules for preliminary tagging, the appropriate tokens are used to fix the correct tags by analyzing the part of the context were tokens are used.

The general form of Brill rule is:

\[ \text{tag1} \rightarrow \text{tag2 if condition} \]

The unknown tokens are tagged initially with the most general parts-of-speech tags. Assigning the appropriate tag depends the on rules that belongs to that context[3][16]. Brill tagger supports specific set of rules for the unknown token similar to the contextual rules.

Fig: 5.2.3.1 illustrates the flow of rule based Parts-of-speech Tagger. The POS tagger receives the strings of text and allocates the Parts-Of-Speech Tags to the tokens of the given strings[68]. The POS taggers are classified in to eight basic group like Verbs, Nouns, Prepositions, etc., The category feature is added to the each and every tokens. The Appendix -II is listed with possible Parts-Of-Speech Tags.

5.2.3.2 CHUNKING

The next process in the pipeline after the Part-Of-Speech tagging is chunking. The Noun phrase Chunker with rule based chunker grammar carries out the task. Each and every noun phrases available
in the given text will be identified by the Noun Phrase chunker[73]. The work of Ramshaw and Marcus on chunking is the foundation of the chunking work.

**Fig: 5.4.3.1 Rule Based Part-Of-Speech Tagger**

To identify the word in the given noun phrase the three tags were used. The three tags are namely I, O and B.

- 'I' is to indicate that the word is inside the Noun Phrase.
- 'O' is to indicate that the word occurrences outside the noun phrase.
- 'B' is to indicate that the word resides between two noun phrases.
Algorithm for Noun Phrase chunking

**Step 1:**
Define a chunk grammar consisting of rules that specify how sentences should be chunked. Illustrate a simple grammar with a regular-expression

(grammar = “NP: {<DT>?<JJ>*<NN>}”)

**Step 2:**
NP chunk should be formed whenever the chunker finds an optional determiner (DT) followed by any number of adjectives (JJ) and then a noun (NN).

**Step 3:**
Test it on POS Tagged sentences.

**Step 4:** Print or display the tree graphically

Chunking is the process of merging the sequence of tokens into syntactic groups like noun phrase group and verb phrase group using the Part-Of-Speech Information’s. It is a beginning phase in the process of parsing a sentence[15][72]. To reach the high accuracy in the biomedical corpora the annotated trained corpus is used. The output of this phase visualized in the form of tree structure.
5.2.4 SEMANTIC ANNOTATION

The study aims to identify the biomedical entities from the available biomedical corpus of text. The process of affixing more information about the notion is known as semantic annotation.

There are various approaches available to identify the biomedical entities include:

- Dictionary Based Approach
- Rule Based Approach
- Machine Learning Based Approach

5.2.4.1 DICTIONARY BASED APPROACH

The collection of huge entity list is known as dictionaries. The entities from the given text will be matched against the dictionary
entries. This method is very specific and known to be Named Entity Recognition method[18]. But the system can only able to identify the entities which are known and available in the dictionary. Unknown and new entities cannot be identified Dictionary based Named Entity Recognition system gives very low recall. Many researchers employ inaccurate matching techniques and produce usual spelling alternative for every entity and include them to the currently available dictionary[22]. Then the updated new dictionary will be used to identify the accurate matches towards the entities in the text.

The major advantages of dictionary-based technique are that it permits to identify entity names and recognizing exclusive concept identities. Matching the entity against the dictionary entries is the simplest approach. This yields very low recall due to embedded features. The biomedical terms varies with the syntactic, semantic and morphological features, not possible to identifying the entities with diverse features using dictionary-based approach.

Dictionary-based approach for biomedical text have two most important performance issues: (i) the biomedical terms have abbreviations and special notations which will considerably degrade the overall accuracy. Elimination of the abbreviations and notations from the dictionary solve this bottleneck. In biomedical text, this kind of solutions will not recognize proper protein or gene names from the
biomedical text. It generates the false positives. (ii) The spelling
difference creates dictionary-based approaches not much usable. In
many situations abbreviations may replace the full text. The dictionary-
based approach identifies the terms using the accurate matching
methods which identifies the abbreviations and full names as distinct
ones.

5.2.4.2 RULE BASED APPROACH

Rule-based approach utilizes the lexical and linguistic rules that
were used to identify the entities. Initially hand crafted rules were used
to recognize the entities from the given text[3][45]. The human experts
involved in crafting the rules. The rules spell out the context and
characteristics of the entities to be identified. The rules can be
expanded further using the grammatical information offered by the
Parts-Of-Speech tagger.

The rule based Named Entity System is not robust. The system
cannot identify the unknown and unseen entities[6]. As a result the
research in the field of biomedical increases leads to inventions of
unknown and unseen entities. And also the process of creating hand
crafted rules is laborious and time consuming. The precision and recall
measures of the system is based on the rules. If the rules are too
specific to the situation then the system yields very good precision and
low recall[62].
5.2.4.3 MACHINE LEARNING BASED APPROACH

Machine Learning based Named Entity Recognition System works better than the dictionary based approach and rule based approach. The machine learning approach uses the general rules, statistical data and algorithms to identify the entities. The Machine Learning based approach produces a model, which is a combination of sequence of various tasks[40][17][32]. The model is used to predict the known and unknowns entities from the given text. The paradigm of machine learning consist of two phases-viz., (i). Training Phase and (ii). Testing Phase. The model have been built using the properties of learning sets and the respective values of the properties from the training data. These kinds of properties are called features. The Machine Learning model uses these features to identify and classify the identified entities in to various classes[41][64]. The Machine Learning model decides whether the entities belongs to the particular class or not.[17] Most of the Machine Learning Models try to identify more collection of features, which is used to distinguish the entities from one class to the other. The Machine Learning based Named Entity Recognition system combines various steps to diverse processing pipelines.
The Machine Learning approach will have training phase and testing phase. In the training phase the documents with annotations is given to the machine learning system[10][40]. The machine learning system using the features and machine learning algorithms will annotate the unannotated documents in the testing phase. The biomedical text is concerned machine learning approach out performs the dictionary based approach and rule based approach[1][33].
5.2.5 Proposed NER Approach

The proposed Named Entity recognition is Feature space based CRF Named Entity Recognition system. The system incorporated with a new hybrid approach which includes two stages namely:

- L-NER (Lexical Named Entity Recognition)
- M-NER (Machine Learning Named Entity Recognition)

The initial stage includes the Lexical based and rule based resources and the later one deals with machine learning algorithms[60]. The conceptual formation can be summarized in two major pipelines.

5.2.5.1. L-NER

The preliminary process in the proposed Named Entity recognition will be L-NER that is Lexical Named Entity Recognition. This phase includes various lexical resources to identify the named entities from the given text. The lexical resources includes:

- Dictionaries
- Lookup tables
- Rule sets
- Ontology
- Lexical words

The lexical based system utilizes the above mentioned lexical resources to identify the known entities from the biomedical documents.
**Dictionaries:** The dictionaries contain a set of lexical words and known entities. Simple rule sets are used to match and extract the entities from the biomedical documents.[15][35][64] Various predefined dictionaries for the field of biomedical technologies are available. The predefined dictionaries are available in the files with extensions ".csv".

**Lookup tables:** One of the important lexical resource is lookup tables. The system loads predefined Lookup tables. The lookup tables hold various predefined biological classes[32]. Normally the predefined lookup tables will be available in the files with extensions ".tsv". The dictionaries can be merged with lookup tables.

**Rule set:** The rule set implies regular expressions based rules. The rules developed using the Per module "Text:: RewriteRule". The process attempts to match the entries of dictionary entries. The system verify for various phrase variations like apostrophe and hyphen. It excludes very tiny phrases. Patterns are integrated to the previously unknown phrases and phrase variants. The pattern templates will be like "([a-z][3;\d]+[A-Z]=\d)". It indicates the phrases that include three lowercase letters followed by uppercase letters and a sequence of zero or more digits.

The Named Entity Recognition may be continued by the lexical resources. The system utilizes the relational expression based text
rewriter to develop and implement the rule sets to access the lexical resources[40][49]. Currently the system identifies 7 classes of biological entities such as:

- Protein
- Gene
- DNA
- RNA
- Cell Line
- Cell Type

from the given biomedical documents which are preprocessed by various processes using diverse Natural Language processing tools[23].

5.2.5.2 M-NER

The M-NER is known as machine leaning based Named Entity Recognition System. Both L-NER and M-NER are responsible for Semantic Annotation. The M-NER uses machine learning algorithm to identify the Named Entities. The L-NER identifies the known entities using the Lexical Resources like dictionaries, rule set, lookup table, ontology etc.. The entities identified in the L-NER are used to train the M-NER in finding the unknown and unseen entities.

5.2.6 Machine Learning Algorithms for Bio-NER

Machine Learning involves in structuring of algorithms. That machine Learning predicts and learns from the data. The machine
learning system constructs the model to predict the output from the given sample inputs.

The three key techniques of leaning Named Entities are:

- Supervised Learning
- Semi-Supervised Learning
- Unsupervised Learning

The supervised learning approach for Named Entity Recognition requires huge amount of annotated corpus as training data. The semi-supervised machine learning system also known as weak supervised learning. It needs training data only in the initial process and then the system starts learning based on the context. The clustering is the best example of unsupervised learning. The Named Entity Recognition System developed based on the concept of Supervised Learning. Various supervised models have been used on NER systems, namely

- Support Vector Machines (SVM)
- Hidden Markov Models (HMM)
- Maximum Entropy Markov Models (MEMM)
- Conditional Random Fields (CRF)
5.2.6.1 SUPPORT VECTOR MACHINES (SVM)

The Support Vector Machine is a supervised machine learning algorithm. It works well for classification problems and regression problems. Vapnik-Chervonenk theory is the base for support vector machines algorithm. SVM positions hyperplane on the feature space and categorize points in that space.

The flexibility is more in Support Vector Machines in selecting similar functions. Support Vector Machines is the only machine learning algorithm used to separate the hyper planes[4]. This learning system has the ability to handle large set of feature space. The soft merging approach of support vector machines control the outfitting problem.

Fig: 5.4.6.1 Support Vector Machine
Hyper plane finding is the most important operation in Support Vector Machine. To train the given data it gives the largest minimum distance. The process of the SVM algorithm is depends on identifying the hyper plane which gives the largest minimum distance to the training data. Twofold, this distance obtains the significant name of margin within SVM’s theory. Therefore, the optimal sorting out of hyper plane maximizes the border of the training data.

5.2.6.2 Hidden Markov Models (HMM)

The statistical Model in which the system assumed to be is modeled in to the Markov process with hidden states is known as Hidden Markov model. It is the simplest form of dynamic Bayesian network.

L. E. Baum and coworkers developed the Hidden Markov model. The applications of Hidden Markov model are temporal pattern recognition system, speech recognition system, handwriting recognition system, gesture recognition system, parts-of-speech tagging system and bioinformatics.

In a Hidden Markov Model, the output of the system depends on the state of the system , which is not directly visible. The Hidden Markov Mode creates the tokens in sequence[8]. These tokens have a probability distribution for every state. Information about the sequence of states will be given by the sequence of tokens. The phrase
'hidden' refers to the sequence of state in which the model passes the parameters of the model.

5.2.6.3 Maximum Entropy Markov Models (MEMM)

The Maximum Entropy Markov Model (MEMM) is a machine learning algorithm which combines the qualities of Hidden Markov Model (HMM). It is a graphical, sequence labeling and discriminative model[24]. The applications of Maximum Entropy Markov Model (MEMM) includes: Information Extraction, Parts-of-speech tagging and Natural Language Processing. In sequence tagging the MEMM out performs the HMM.[29] To define the special purpose feature, the domain knowledge is used in the Maximum Entropy Markov Model (MEMM). The training phase of this machine learning algorithm is more powerful than Hidden Markov Model (HMM). But Maximum Entropy Markov Model (MEMM) algorithm suffers from “label bias problem” which is a major disadvantage to this algorithm[21].

5.2.6.4 Conditional Random Fields (CRF)

Conditional Random Fields machine learning algorithm’s are discriminative undirected probabilistic graphical model[38]. The known and unknown relations between interpretations are trained using the machine learning algorithm Conditional Random Field[29]. The applications of Conditional Random Fields algorithm’s are identifying biological sequences, shallow parsing and named entity recognition
Another major domain area of Conditional Random Fields application include object recognition and image segmentation.

The Conditional Random Fields (CRFs) are undirected graphical models, based on conditionally-trained finite state machines[48]. The major gain of Conditional Random Fields is the provision to incorporate a large variety of random features and non-independent features of the input[59]. Automated feature initiation allows to enhanced accuracy and also huge decrease in parameters. The process implements the linear-chain CRFs.

Suppose \( P(Y_v \mid X, \text{all other } Y) = P(Y_v \mid X, \text{neighbors}(Y_v)) \)
then \( X \) with \( Y \) is a conditional random field

- \( X \): observations, \( Y \): labels
- \( P(Y_3 \mid X, \text{all other } Y) = P(Y_3 \mid X, Y_2, Y_4) \)

![Conditional Random Fields Diagram](image)

### 5.2.7 PROPOSED MACHINE LEARNING ALGORITHM FOR BIO-NER

Identifying biological named entities from the biomedical document is a challenging task[34]. The proposed method of bio Named Entity recognition includes two phases namely L-NER and M-NER. The
L-NER represents the Lexical based Named entity Recognition whereas M-NER represents the Machine Learning based Named entity Recognition[57][76]. The L-NER utilizes the lexical resources like dictionaries, Rule set, ontology's, lookup tables to identify the Named Entities. And in the second phase the M-NER that is Machine Learning Named Entity Recognition system employ machine learning algorithm to identify the known and unknown entities from the given documents[71]. There exists a wide variety of machine learning algorithms, but the Conditional Random Field algorithm out performs the other machine learning algorithms like Support Vector Machines (SVM), Hidden Markov Models (HMM), Maximum Entropy Markov Models (MEMM). CRF has several advantages over other methods.

- CRF avoid the label bias problem which is a weakness of Maximum Entropy Markov Models (MEMM).

- The conditional nature of Hidden Markov Models (HMM) results in relaxation of the independence assumptions which can be overruled by CRF

- Support Vector Machines (SVM) required more time to train complex models than CRF.

The proposed system "Feature based Bio NER Recognition System" comprises Feature space based Conditional Random Field to recognize the entity from the document[39][65]. Feature space based
Conditional Random Fields (CRFs) are undirected statistical graphical models, that corresponds to conditionally trained finite-state machines well suited for labeling and segmenting sequence data. Named entity recognition can be framed as a sequence labeling problem.

5.2.7.1 FEATURE SPACE GENERATION

Feature is a numerical representation of raw data. The raw data is represented in a high dimensional vector. The feature extraction initiates from a preliminary group of quantified data and constructs values from that called features[61]. An algorithm provided with huge set of data to process where the data may be redundant, in that situations it may be reduced to a minimal set which is known as feature.

The definition of a rich and carefully selected set of features is required in order to properly represent the target entity names.

The Feature Space is based on

- Orthographic Features
- Morphological Features
- Context Features
- Lexicons Features

The Feature selection is deriving a subset of the feature set. The selected information from the process of feature selection should contain related information from the initial data than the specified task.
can be done using the selected features instead of using the whole preliminary data. The feature space generation module is to generate features and to create matrix of prediction during the recognition process in the Machine Learning phase[3][50]. The features include various Natural Language Processing components such as dictionary entries, hand crafted and automated rule set, POS tagger, lemmas, noun phrase chunkers, shallow parser and dependency parser etc.. In the training model the features are generated for the word tokens and sentences[3][43][61]. The whole process were structured in two major pipelines such as : The initial structure creates a model from the annotated corpus in the L-NER (Lexical Named Entity Recognition) phase. The second phase uses the Feature space based Conditional Random Field Algorithm which is a machine learning algorithm to identify the known and unknown entities.

Words in a sentence are tokens to be assigned labels by states in the CRF framework.

Let \( o = \langle o_1, o_2, o_3, \ldots, o_n \rangle \) be a sequence of observed words of length ‘n’

Let \( L = \) (protein, DNA, other, etc.) be a set of labels corresponding to states in a finite-state machine.

Let \( l = \langle l_1, l_2, l_3, \ldots, l_n \rangle \) is a sequence of labels from ‘L’ assigned to words in the input sequence ‘o’.
A first-order linear-chain CRF defines the conditional probability of a label sequence given an input sequence to be:

\[
P(l|o) = \frac{1}{Z_0} \exp \left( \sum_{i=1}^{a} \sum_{j=1}^{k} \lambda_{ij} f_j(l_{i-1}, l_i, o, i) \right)
\]

Where

- \(Z_0\) : Normalization factor over all possible label sequences.
- \(f_j\) : one of the \(k\) binary functions describing a feature at position \(I\) in sequence \(o\) and \(\lambda\) is a weight for that feature.

**5.2.7.2 FEATURE SPACE BASED CRF IMPLEMENTATION**

The Feature space based Conditional Random Field algorithm is implemented with the MALLET toolkit. MALLET uses a quasi-Newton method called L-BFGS (Limited-memory Broyden–Fletcher–Goldfarb–Shanno) to find the optimal feature weights efficiently[61]. The system is developed entirely in Java using graphical window objects from the Swing library.

The performance results are obtained using three important measures such as:

- Precision
- Recall
- F-Measure
Based on the above mentioned three factors, the system evaluates and compares with Support Vector Machines Algorithm using data from three different corpus such as Breast Cancer Corpus, Lung Cancer Corpus and Thyroid Cancer Corpus[33]. The system identifies seven classes of entities from the given corpuses. Based on the outcomes of the analysis, it is very clear that the system out performs the existing systems in many aspects.