Chapter 4

Machine Learning based NER for Assamese

4.1 Introduction

For a resource-rich language like English, NER has shown high accuracies. But development of NER for a resource-poor language like Assamese is challenging due to unavailability of proper resources. In this chapter we present work on NER using CRFs and HMM. Our's is the first such work on NER in Assamese using CRFs and HMM. The ML approach has advantages over the rule-based approach in that it is adaptable to different domains, has robustness, and moreover, it is not a time-consuming process, whereas the rule-based approach is labor-intensive and time-consuming. Hence, we prefer to use an ML approach, i.e., CRFs and HMM, which results in an accuracy of 75%-85%. Finally, we also propose a hybrid approach which shows an improvement over both CRF and HMM.

4.2 Features used in Named Entity Recognition

Different features of words can help in NER. For example, the Boolean feature of capitalization provides a clue to NE:- if a word is capitalized it is represented as true, otherwise it is false. Nadeau and Sekine [87] present three types of attributes that an NER system may represent. We give examples of each type below.

1. Capitalization or lack of it can be represented by a Boolean attribute.
2. The length of a word in terms of characters can be represented as a numeric attribute.
3. The lower case version of a word can be represented as a nominal attribute.

For example, *The President of India attended the conference* excluding the punctuation would be represented by the following feature vector:
<true, 3, "the">, <false, 9, "president">, <false, 2, "of">, <true, 5, "india">, <false, 8, attended>, <false, 3, "the">, <false, 10, "conference">.

Different types of contextual information along with a variety of other features are used to identify NEs. Prefixes and suffixes of word also play important roles in NE. The features used may be language-independent or dependent. Different language independent features that help in identifying NEs include contextual information, prefixes and suffixes of all the words, NE tags of previous and following word(s), whether it is the first word, length of the word, whether the current "word" is a sequence of digits, whether the current word is infrequent and the POS of the current and surrounding words . In contrast, language dependent features include the set of known suffixes; clue words that help identify person, location and organization names; and designation words that help to identify person names which is described below.

Language independent features used in NER include the following.

1. *NE information*: The NE tag information for the previous and the following words are important features in deciding the NE tag of the current word. For example, ৰাম অসমলৈ গল [ɾam ɒsɒmoloi gol] [E:Ram went to Assam]. In this example, ৰাম [ɾam] is a person NE which helps identify that the next word is likely to be again an NE. Similarly in Bengali, ৰাম আছামে গিয়েছিল [ɾam asame giyesil] [E:Ram went to Assam] can also help to identify the person NE.

2. *Digit features*: Different types of digit features have been used in NER. These include whether the current token is a two-digit or four digit number, or a combination of digits and periods and so on. For example, ৫ জুন ২০১১ [5 jun 2011].

3. *Organization suffix word list*: Several known suffixes are used for organizations. These help identify organization names. For example, if there exists a word like *Ltd* or *Co*, চৰকাৰ [sɒɾkaɾ] [E:Govt] it is likely to be a part of an organization's name.

4. *Length of words*: It is often seen that short words less than 3 characters are not usually NE. But there are exceptions, e.g., ৰাম [ɾam] [E:Ram], সীতা [sita] [E:Sita], ৰণ [ɾon] [E:Ron].

Language dependent features used in NER include the following:

1. *Action verb list*: Person names generally appear before action verbs. Examples of such action verbs in Assamese are কৈছিল [koisil] [E:told], গৈছিল [gɔisil] [E:Went]. কথাটো ৰামে কৈছিল [kot^{h}atu rame koisil] [E:Ram told it].
সিহতৰ ঘৰত হৰি গৈছিল [shihotor g^{h}ɔrot hɔri gɔisil] [E:Hari went to their home].

2. *Organization suffix word list*: It also acts as a language dependent features

such as in Assamese there are some suffixes used for organization names such as গোট [gɒt] [E:Group] which identify an organization.

3. *Context word features*: Surrounding words, such as the previous and the next word of a particular word serve as important features when finding NEs. For example, a word like জিলা [zila], পুৰ [puɾ] or পাৰা [paɾa] indicates the presence of a location. These words are used to identify location names. Similarly, ওস্তাদ [ustad] [E:Expert], ক্ৰীড়াবিদ [kriɾabid] [E:Sportsman] and কবি [kɒbi] [E:Poet] denote that the next word is a person name.

4. *Word prefix and suffix*: A fixed-length prefix or suffix of a word may be used as a feature. It has been seen that many NEs share common prefix or suffix strings which help identify them. For example, in Assamese দাদা [dada] [E:Older Brother], বাইদেউ [baidɛu] [E:Older Sister] are used identify person NEs. Similarly in Bengali, দাদা [dada] [E:Older Brother], দিদি [didi] [E:Older Sister] are used to identify a person NEs.

5. *POS*: Part-of speech is an important feature in identifying the NEs. For example, if two words in sequence are both verbs, the previous word is most likely to be a person name. Example: কমল দৌৰি আহিছে [kɒmɒl dɒuɾi ahise] [E:Kamal came running]. Similarly in Bengali we can say as কমল খেয়ে ঘোমাইছে [kɒmɒl kʰeye gʰumaise] [E:Kamal slept soon after having food].

6. *Designation words*: Words like Dr, Prof etc often indicate the position and occupation of named persons, serving as clues to detect person NEs. For example, in Assamese we can say as প্ৰফেচৰ দাস [profɛsor dɑs] [E:Professor Das], মন্ত্ৰী বৰাই কয় [mɒntɾi bɒɾai koi] [E:Minister Bora said].

4.3 CRF Approach

CRFs are a type of discriminative probabilistic model used for labeling and segmenting sequential data such as natural-language text or biological sequences. CRFs represent an undirected graphical model that define a single non-linear distribution over the joint probability of an entire label sequence given a particular observation sequence. CRFs can incorporate a large number of arbitrary, non-independent features and are used to calculate the conditional probabilities of values on designated output nodes, given the values on designated input nodes.

Lafferty et al.[69]define the the probability of a particular label sequence y given observation sequence x to be a normalized product of potential functions, each of the form

$$\exp(\sum_{j} \lambda_j t_j(y_{i-1}, y_i, x, i) + \sum_{k} \mu_k s_k(y_i, x, i)) \tag{4.1}$$

where $t_j(y_{i-1}, y_i, x, i)$ is a transition feature function of the entire observation sequence and the labels at positions i and $i-1$ in the label sequence; $s_k(yi, x, i)$ is a state feature function of the label at position i and the observation sequence; and λ_j and μ_k are parameters to be estimated from training data.

When defining feature functions, one constructs a set of real-valued features $b(x, i)$ of the observation to expresses some characteristic of the empirical distribution of the training data that should also hold of the model distribution. An example of such a feature is

$$b(x, i) = \begin{cases} 1 & \text{if the observation at position } i \text{ is the word "September"} \\ 0 & \text{otherwise,} \end{cases}$$

Each feature function takes on the value of one of these real-valued observation

features $b(x, i)$ if the current state (in the case of a state function) or previous and current states (in the case of a transition function) take on particular values. All feature functions are therefore real-valued. For example, consider the following transition function:

$$t_j(y_{i-1}, y_i, x, i) = \begin{cases} b(x, i) & \text{if } y_{i-1} = \text{IN and } y_i = \text{NNP} \\ 0 & \text{otherwise,} \end{cases}$$

The notation is simplified by writing
$s(y_i, x, i) = s(y_{i-1}, y_i, x, i)$ and $F_{j(y,x)} = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i)$,

where each $f_j(y_{i-1}, y_i, x, i)$ is either a state function $s(y_{i-1}, y_i, x, i)$ or a transition function $t(y_{i-1}, y_i, x, i)$. This allows the probability of a label sequence y given an observation sequence x to be written as

$$p(y|x, \lambda) = \frac{1}{Z(x)} \exp(\sum_j \lambda_j F_j(y, x)). \tag{4.2}$$

$Z(x)$ is a normalization factor.

4.3.1 Experiment

When applying CRFs to the NER problem, an observation sequence is of tokens of a sentence or document of text and the state sequence is the corresponding label sequence. We have used the library called Stanford NER, which is a simple, customizable, and open-source Java implementation of CRF for segmenting or labeling sequential data. In the supervised ML approach, labeled training data act as essential inputs to calculate the probability of a tag to be assigned to a word in an untagged corpus. It is necessary to obtain quality training data for supervised learning to be successful. The size of the training data also must be

large enough for effective learning. Since NER is the process of identification and classification of the proper noun into different classes, the data must be annotated with appropriate predefined labels. For the Stanford NER, the training file should be in a tab-separated column, i.e., words in column 0 and the corresponding label in column 1. For this purpose our corpus is tokenized into a word per line and is annotated with the three labels, viz., person, location, and organization.

For example, consider a sentence হীৰেন দাসে মহাত্মা গান্ধী ৰোড হৈ গৈছে। প্রতিদিন সংবাদ গুবাহাতী এক অনুষ্ঠানত.

Thus the format of the training file for the above example is shown in the Table 4.1.

Annotation must be carried out by a domain expert to ensure quality. Maintaining the quality of the training data is a difficult task when several human experts are engaged in labeling, as variations and inconsistencies may show up. These variations exist as differences arise among human experts while performing the annotation task. The Inter-Annotator Agreement (IAA), also known as Inter-Annotator Consistency is a widely used term in annotation. The main goal of this agreement is to identify how well different annotators agree in the same annotation process for the defined classes [127]. To reach a high level of IAA, multiple annotators must work in the same annotation task in an iterative way so that all the discrepancies can be identified and the best output produced. This overall process is time-consuming. However, in most cases discrepancies still exist even if a rigorous process is followed [84]. The unannotated data are the test data. An example of an annotated sentence is given below:

> <ENAMEX TYPE="PERSON"> MR. X </ENAMEX> visited <ENAMEX TYPE="LOCATION"> U.S </ENAMEX> to attend a conference held in <ENAMEX TYPE="ORGANIZATION"> ABC Ltd. </ENAMEX>.

Table 4.1: Example of training file for CRF

Word	Tag
হীৰেন	Person
দাস	Person
মহাত্মা	Person
গান্ধী	Person
ৰোড	Location
হৈ	0
গৈছে	0
প্ৰতিদিন	0
সংবাদ	0
গুবাহাতী	Location
এক	0
অনুষ্ঠানত	Organization

A considerable amount of work is seen in English, German and Chinese using the ML approach, and the use of the ML approach is common in Indian languages. Indian languages also suffer from lack of annotated resources compared to other languages. For our NER task, articles from the reputed Assamese newspaper *Asomiya Pratidin* and the *Emille Corpus* described in Chapter3 consisting of 0.2 million wordforms are used. Our corpus is split into two sets, one forms the training data, and the other forms the test data. The CRFs are trained with the training data and the test data is tagged using the CRF model. We have prepared the training data manually by annotating with different classes of NEs. Issues that arise in annotating the data which are as follows:

- Consider a NE like (মহাত্মা গান্ধী ৰোড)[mahatma gand^{h}i rɒd] [E:Mahatma Gandhi Road]. Here (মহাত্মা)(গান্ধী) [E:Mahatma Gandhi] is tagged as a person and (মহাত্মা গান্ধী ৰোড) [E:Mahatma Gandhi Road] as a location. In other words, the sequence of the three words should be treated as a location name. In general, this is a difficult issue in NER.

- Words like (অন্ধ্র-প্রদেশ) [and^{h}ra pradesh] [E:Andhra-Pradesh] should be treated as a single NE. While preparing the training file, whenever a token like (অন্ধ্র-প্রদেশ) [and^{h}ra-pradesh] [E:Andhra-Pradesh], or (ড ঃ) [E:Dr :] is found, it is treated as several separate words as অন্ধ্র [E:Andhra], - , প্রদেশ [E:Pradesh] and ড [E:Dr], ঃ [E: (:)] according to the CRF rules.

- Spelling variation of particular words like (সিং) [singh] and (সিঙ) [singh] also cause problems.

  When preparing the training file in order to work with CRFs, we need to create a serialized file that stores the probability of a particular order of training data. It is then used to find the proper tag sequence for a given word sequence and to tag the test data with the three defined NE classes. Table 4.2 shows the format

of the serialized file for the CRF classifier. The first column is the number of iterations, LINESEARCH is used to determine the maximum amount to move along a given search direction, VALUE is the probability function value, TIME is the total elapsed time required to calculate the probability, GNORM is the normalization factor, and finally the AVEIMPROVE is the average improvement/current value after normalization.

Table 4.2: Format of the Serialized file

Iter	[LINESEARCH]	VALUE	TIME	GNORM	AVEIMPROVE
Iter 1	[11M 1.008E-4]	1.456E2	0.10s	—1.600E2—	0.000E0
Iter 2	[33M 2.100E1]	1.066E2	0.19s	—5.772E1—	1.832E-1
Iter 3	[M 1.000E0]	1.014E2	0.24s	—8.977E1—	1.453E-1
Iter 4	[M 1.000E0]	8.994E1	0.27s	—2.693E1—	1.547E-1
Iter 5	[M 1.000E0]	8.876E1	0.30s	—1.326E1—	1.281E-1
Iter 6	[M 1.000E0]	8.712E1	0.33s	—1.029E1—	1.119E-1

4.3.2 Results and Discussion

We conducted standard 3-fold experiments. In each fold, there are training data and test data. Then in each fold, a learning model is created based on the training data. Out of .2 million wordforms, a set of 130K wordforms have been manually tagged with four tags namely person, location, organization and miscellaneous. This set is used as the training set for the CRF based NER system and the remaining 70K wordforms are considered test data. The words which were unseen during the training phase are assigned the class 0. We

present the precision, recall and f-measure for each of the 3-fold experiments in Table 4.3, Table 4.4 and Table 4.14.

Table 4.3: NER results for Set 1 using CRF

Classes	Precision	Recall	**F-measure(%)**
Person	83.8	75.6	79.4
Location	78.5	85.9	82.03
Organization	79.7	83.3	81.4
Miscellaneous	80.5	78.4	79.4

Table 4.4: NER results for Set 2 using CRF

Classes	Precision	Recall	**F-measure(%)**
Person	97	62	75.6
Location	91.7	60.6	72.9
Organization	84	76.2	79.8
Miscellaneous	85	72.8	72.8

We see that our CRF-based NER system encounters errors while labeling the NEs. We use different language-dependent and independent features, but sometimes wrong labels assigned. Examples of such errors are: (মূখ্যমন্ত্রী পত্নী হোবাৰ) [mujɒmɒntɾi pɒtni huwar] [E:being the wife of Chief Minister], (ৰাজ্যপাল শ্রীনিবাস) [ɾajɒpal sɾinibas] [E:Governor Srinivas] , (আঠগৰাকী সাহিত্যিক আছিল) [aatʰiɡoɾaki sahitŋk aasil] [E: There were eight witers]. Whenever the system finds words like মূখ্যমন্ত্রী [mujɒmɒntɾi], ৰাজ্যপাল [ɾajɒpal] and সাহিত্যিক [sahitŋk], the next word is tagged as a person name when often they are not.

Table 4.5: NER results for Set 3 using CRF

Classes	Precision	Recall	**F-measure(%)**
Person	79.5	81.2	80.2
Location	80	82.1	81.03
Organization	78.4	79.5	78.9
Miscellaneous	83.2	85.4	84.2

Table 4.6: Average CRF Results

Classes	**F-measure(%)**
Person	78.4
Location	78.65
Organization	80.03
Miscellaneous	78.8

So, in such cases more careful rules need to be derived in order to avoid these errors. As discussed in the training file words like (ড ঃ) [E:Dr :] are considered as two separate words, giving a wrong tag. While considering the digit features, whenever a comma, or a colon is found with a date and a time, our system gives a wrong tag. For example, (১ জুন, ২০১২) [E:1 June 2012], (৩-৪ মে) [E:3-4 May].

The only way to avoid these errors is to explore additional features besides the ones we have used. Another way to improve the performance of the system is to increase the size of the training file and to explore some more features for each class.

4.4 The HMM Approach

An HMM is a statistical model that can be used to solve classification problems that have an inherent state sequence representation. The model can be visualized as a collection of states. These states are connected by a set of transition probabilities that indicate the probability of traveling between two given states. A process begins in some state, and moves through new states as dictated by the transition probabilities. In an HMM, the exact sequence of states that the process generates is unknown (i.e., hidden), hence it is a hidden model. The output of the HMM is a sequence of output symbols. A Markov chain assumes that the probability of a tag being the next state depends on the previous tag. For example, consider the sentence:
Ram is playing cricket.
In this sentence after the verb *playing*, it is most likely that the next word will be a noun or preposition, which is dependent on the previous tag. Markov Model is used to find the highest probability of a particular tag sequence for a given word sequence. NER may be viewed as a classification problem, where every word is either part of some name or not part of any name. The bigram statistical model is used to obtain the NCs (Name Class), which are dependent on the previous words. For our purpose of name-finding, given a sequence of words (W), we need to find the most likely sequence of Name Class(NC)Bikel et al.[74] ,i.e.,

$$max \Pr(NC|W) \tag{4.3}$$

By using Bayes theorem,

$$\Pr(NC|W) = \frac{\Pr(W, NC)}{\Pr(W)}. \tag{4.4}$$

Now, as Pr(W), the unconditioned probability of word sequence, does not change as we consider various values of NC, our main aim is to maximize the numerator, i.e., the joint probability of the word sequence and the name-class sequence. The HMM approach for attaining this joint probability is based on the below three components.

- State transition probabilities, i.e., the probabilities of moving from one state to another state. In case of NER, it means moving from one NC to another NC, e.g. from Person to Location, Person to Person, Location to Person etc. The current named class probability is conditioned on the previous named class and the previous word.

$$\Pr(NC|NC_{-1}, w_{-1}) \tag{4.5}$$

- Probability of generating the first word inside a name class. The first word is generated based on the current and the previous name-class, also known as initial probability.

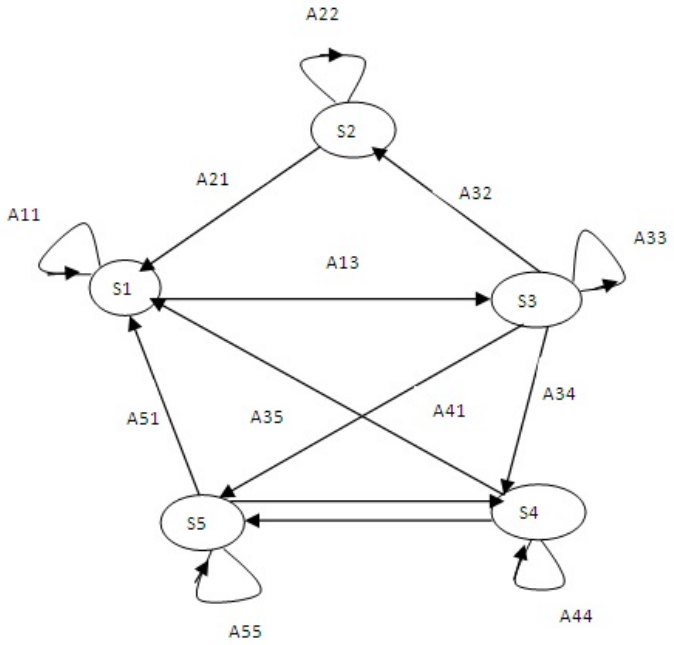
$$\Pr(< w, f >_{first} |NC, NC_{-1}). \tag{4.6}$$

- Observation probabilities, i.e., a model to generate all subsequent words within the name-class. Subsequent words are generated based on the immediate predecessor and current name class:

$$\Pr(< w, f > | < w, f >_{-1}, NC). \tag{4.7}$$

  where w = word, NC = name-class, NC_{-1} = previous name-class, w_{-1} = previous word.

Figure 4.1: State Transition Diagram for states S1,S2,S3,S4 S5



These three steps are repeated until the entire observed word sequence is covered. We search the entire space of all possible name-class assignments, using the Viterbi algorithm (Viterbi, 1967), and maximizing the numerator Pr(W, NC).

A Markov state transition diagram for 5 different states (S1, S2, S3, S4, S5) is shown in Fig 4.1 where (Aij) represents the transition from one state to another.

4.4.1 Our Experiment

To implement the HMM model, we prepared a training file of 0.15 million wordforms. The training file is prepared manually by annotating the data

with different classes of NEs. The format of the training file is as follows.

হেমন্ত<person> দাসৰ<person> দ্বাৰা<other> পৰিচালিত<other>। [ɦɛmɒnto dasɒr dara porisalito] [E:Directed by Hemanta Das].

যিসকলে<other> গৰিমাৰে<other> দেশে<other> । [jisokol gɒrimar desʱɛ] [E: With pride for the country].

অসমত<location> যিমান<other> সম্পদ<other> আছে<other> । [ɒsɒmot jiman sɒmpod aase] [E: Resources available in Assam].

When calculating the probabilities of Equations (4.6) and (4.7), we consider the word features associated with each word. Word features play an important role in identifying NEs. The features that we use in our experiment are the same as those used in the CRF approach, which is discussed in Section 4.2. The calculation of the above probabilities is straightforward, using events/sample-size.

– Probability defined in Equation 4.3 is calculated using the formula below:

$$\Pr(NC|NC_{-1}, w_{-1}) = \frac{c(NC, NC_{-1}, w_{-1})}{c(NC_{-1}, w_{-1})}. \tag{4.8}$$

– Probability defined in Equation 4.4 is computed using the formula below:

$$\Pr(< w, f >_{first} |NC, NC_{-1}) = \frac{c(< w, f >_{first}, NC, NC_{-1})}{c(NC, NC_{-1})}. \tag{4.9}$$

– Probability defined in Equation 4.5 is calculated using the formula below:

$$\Pr(< w, f > | < w, f >_{-1}, NC) = \frac{c(< w, f >, < w, f >_{-1}, NC)}{(< w, f >_{-1}, NC)}. \tag{4.10}$$

In the above, c() represents the number of times the events occurred in the training data (the count). For example, let us consider the sentence below. (ৰামে ডঃ জানকী ৰৱা বিদ্যালয়ত পঢ়ে) [rame dr: jonaki bɒra bidyalɒit pɒrhe] [E: Ram studies in Dr Janoki Bora School]. Here NC = [person, location, organization, miscellaneous, other] and w = current word.

In the above example, (ৰামে)[E:Ram] is the beginning of the sentence as well as the beginning of a NC. Thus, the initial and the transition probabilities are calculated considering these two factors, i.e., the beginning of a sentence, and beginning of a NC. Here, w_{-1} = ।(daari) and NC_{-1} = start of the sentence. Since both w_{-1} and NC_{-1} are constant (i.e., w_{-1} and NC_{-1} never change for the start of a sentence), the transition probabilities of person, location, organization, miscellaneous and other are calculated considering the count of different NCs with which a sentence begins in the training data. The initial probability is calculated by simply counting how many times the word (ৰামে) [E: Ram] has occured in the training data as the first word of different NCs. The product of the above two probabilities yields the first tag of the desired NC, based on the maximum probability value. Once the initial NC is decided, we need to find the next words within the same NC using Equation (4.8). Thus, if we find (ৰামে) [E: Ram] as person, we need to check whether the word (ডঃ) [E: Dr:] comes under the same name-class. But the observation probability of the word (ডঃ) [E: Dr:] for a person NC is zero because the word feature associated with the word (ডঃ) [E: Dr:], i.e., title, can never be a middle word or the last word of the person NC. Thus (ৰামে) [E: Ram] is be marked as the end of the person NC and we calculate the transition and the initial probabilities for the word (ডঃ) [E: Dr:] for the beginning of a separate NC. This time we have to consider both NC_{-1} and w_{-1}, since these two factors are not constant any more for the middle of a sentence. In this way, we evaluate the

maximum probability of the tag sequence given a word sequence. The correct tag sequence for the above sentence is as follows:

<person>রামে [rame]</person> <org>ডঃ জানকী বরা বিদ্যালয়ত [dr: jonaki bɒra bidyalɒit]</org> <other>পঢ়ে [pɒrhe] </other>.

The larger the training data, the higher the probability of getting the correct results.

4.4.2 Results and Discussion

We have conducted a three-fold experiment. The test data and the training data are the same as used with the CRF experiment discussed in Section 4.3.2. We present the precision, recall and f-measure for each of the conducted 3-fold experiments in Table 4.7, Table 4.8 and Table 4.9

Table 4.7: NER results for Set 1 using HMM

Classes	Precision	Recall	**F-measure(%)**
Person	82	84	84
Location	83.1	81.2	82
Organization	80	83.4	81.6
Miscellaneous	79.2	80	79.5

Below are the major issues encountered when tagging a file with the HMM approach.

1. *Ambiguity in names*: As the same name can be assigned to more than one NCs(ambiguity), issues arise when deciding the correct tag sequence.

Table 4.8: NER results for Set 2 using HMM

Classes	Precision	Recall	**F-measure(%)**
Person	81.2	83	82
Location	83	82	82.4
Organization	78	80	78.8
Miscellaneous	80	79	80.65

Table 4.9: NER results for Set 3 using HMM

Classes	Precision	Recall	**F-measure(%)**
Person	82.4	80	81
Location	83	81.2	82
Organization	81.5	83	78.5
Miscellaneous	82.4	85	83.6

Table 4.10: Average HMM Results

Classes	**F-measure(%)**
Person	82.33
Location	82.13
Organization	79.6
Miscellaneous	81.25

The problem is elaborated with the example below.

Suppose we have the following three sentences in the training file.

(a) ৰামে<person>ৰাধা<org> ৰৱা<org> কলেজত<org> পঢ়ে<other> । [rame radɦa bora kolejot porhe] [E:Ram studies in Radha Bora College].

(b) ৰামে<person>ৰাধা<person> ৰৱা<person> অহা<other> দেখি<other> থিয়<other> হ'ল<other> । [rame radbora ɒha dekɦi tɦiyo hol] [E:Ram stood up seeing Radha Bora].

(c) ৰামে<person>ৰাধা<person> ৰৱা<person> অহা<other> বুলি<other> গম<other> পাই<other> লগ<other> ধৰিবলৈ<other> গ'ল<other> । [rame radbora ɒha buli gom pai log dɦoriboloi gol] [E: Ram went to meet Radha Bora after knowing that he came].

We need to find the tag sequence of the sentence [ৰামে ৰাধা ৰৱা কলেজত পঢ়ে ।] [E:Ram studies in Radha Bora College]. Since the word (ৰাধা) [E:Radha] has no specific word feature associated with it to signify the start of an organization or person NC (unlike ডঃ জানকী ৰৱা) [E:Dr Janaki Bora], and also, since the bigram probability is calculated, (ৰাধা) [E:Radha] will be considered as the next word of the current person NC. So,the complete tag sequence is as given below:

<person>ৰামে ৰাধা ৰৱা [rame radɦa bora] < /person> <org>কলেজত [kolejot]</org> <other>পঢ়ে [porhe]</other> ।

which is incorrect.
The correct tag sequence should be as shown below:

<person>ৰামে [rame]</person> <org>ৰাধা ৰৱা কলেজত [radɦa bora kolejot]</org> <other>পঢ়ে [porhe]</other> ।

2. *Unknown words*: The simple ML approach is unable to handle unknown words, i.e., words not present in the training file. Unknown words may occur in the test data in three different ways in the bigram model.

   - As the current word,
   - As the previous word, and
   - Both as current and previous words.

Smoothing is applied to handle unknown words using the back-off model described below.

- *Named class Back-off*: When calculating the probability $Pr(NC|NC_{-1}, w_{-1})$, if w_{-1} is unknown, we calculate the probability using $Pr(NC|NC_{-1})$ i.e.,

$$\Pr(NC, NC_{-1}) = \frac{c(NC, NC_{-1})}{c(NC_{-1})}. \tag{4.11}$$

- *First word back-off*: If w is not available in the training data as the first word of the desired NC, the back-off model calculates the probability of <w,f> based on only the NC, i.e., the count of <w,f> within the NC.

Thus the results for the three sets of data are shown in Table 4.11, Table 4.12 and Table 4.13.

Thus, we see that after smoothing is applied, there is a decrease in precision whereas the recall increases, which results in a slight improvement in F-measure. This is due to the reason that in smoothing the w_{-1} and <w,f> are ignored, resulting in a increase of the total number of NEs retrieved and the number of correctly retrieved NEs.

Table 4.11: NER results for Set 1 using HMM and Smoothing

Classes	Precision	Recall	**F-measure(%)**
Person	80	90	84.7
Location	81.2	84	82.5
Organization	78.5	85	81.6
Miscellaneous	79.2	80	79.5

Table 4.12: NER results for Set 2 using HMM and Smoothing

Classes	Precision	Recall	**F-measure(%)**
Person	79.3	86	82.5
Location	77	83.4	84.2
Organization	80	81.3	80.6
Miscellaneous	80	79	80.65

Table 4.13: NER results for Set 3 using HMM and Smoothing

Classes	Precision	Recall	**F-measure(%)**
Person	80.1	85	81.5
Location	80	87	83.3
Organization	78	85.2	81.4
Miscellaneous	82.4	85	83.6

We also extend our experiment on two North-Eastern Indian languages namely Bodo and Bishnupriya Manipuri(BPM). BPM is an Indo-Aryan language

Table 4.14: Average HMM Results after applying Smoothing

Classes	**F-measure(%)**
Person	82.9
Location	83.3
Organization	81.2
Miscellaneous	81.25

spoken in parts of Indian states of Assam, Tripura as well as in the Sylhet region of Bangladesh, and in Burma. It is written in Bishnupriya Manipuri script which is almost similar with Bengali and Assamese scripts. The total number of speakers is 4,50,000 approximately. This script has 8 vowels and 25 consonants. On the other hand Bodo is a tonal language with two tones, belonging to the Tibeto-Burman language family. Bodo language is written using Devanagiri script. Bodo is spoken mainly in North-East India and Nepal and is closely related to the Dimasa language of Assam, Garo language of Meghalaya and Kokborok language of Tripura. It has 6 vowels and 16 consonants sounds. The total number of speakers is 1,222,881 according to 1991 census. The HMM based NER system is trained and tested with both Bodo and BPM languages. Corpora of 11K wordforms are used for both BPM and Bodo. The result obtained for both the languages is shown in the Table 4.15 and 4.16 below.

Table 4.15: NER results for Bodo Dataset

Classes	Precision	Recall	**F-measure(%)**
Person	75	50	60
Location	75	25	36
Organization	68	72.1	69.9
Miscellaneous	100	60	75

Table 4.16: NER results for BPM Dataset

Classes	Precision	Recall	**F-measure(%)**
Person	89.7	50	64.2
Location	81.8	32.7	46.7
Organization	100	50	66.6
Miscellaneous	75	50	60

4.5 A Hybrid approach for NER

Hybrid approach is an approach where more than two approaches are used to improve the performance of NER system. We improve the performance of Assamese NER presented earlier in this thesis to some extent by integrating the ML approach with the rule-based and gazetteer-based approaches to develop a hybrid system. To the best of our knowledge, there is no work on hybrid NER in Assamese. We develop a hybrid NER system that has the ability to extract four types of NEs. Each of the approaches has its own strengths and weaknesses. Here, we describe the hybrid architecture, which produces better results than the rule-based approach or ML individually.

The processing goes through three main components: Machine-learning, rule-based, and gazetteer-based. The machine-learning components involve two approaches, CRFs, and HMM. Various NE features are used when implementing the two approaches. The rule-based approach involves the rules that we have derived for different classes of NEs and the gazetteer-based approach involves the tagging of NEs using the look-up lists for location, person, and organization names. We have observe that certain methods are superior in handling certain issues better than other models and vice versa. Thus, we define a precedence of methods to be applied on the output to another. Below are the steps used in our proposed hybrid model.

1. With a large amount of training data, ML approaches normally give better results then other methods when applied individually.

2. Apply ML on the raw test data.

3. The rules for multi-word person names, organization names and location names discussed in Chapter 3 have the best results in dealing with names with clue words. Thus, these rules are applied to the output of the ML approach, and this will not only tag the left-out data but also overwrite the existing tagged data whereever applicable. This will help us in effectively handling the errors encountered in implementing the HMM, i.e., lack of word features and ambiguity in names.

4. Apply the gazetteer-based approach on the untagged data of the output of Step 2.

5. Apply the ML-based smoothing technique on the remaining left-out words.

6. Apply rules for single word person names as discussed in Chapter 3 as the last step on the untagged data.

The overall architecture of our hybrid approach is shown in Fig4.2. The results obtained after applying the hybrid approach are shown in Table 4.17, Table 4.18 and Table 4.19 and the average result is shown in Table 4.20.

Table 4.17: NER results for Set 1 using Hybrid Approach

Classes	Precision	Recall	**F-measure(%)**
Person	87	86.1	86.4
Location	87	83.2	85.05
Organization	88.1	86	86.4
Miscellaneous	90	88	88.8

Table 4.18: NER results for Set 2 using Hybrid Approach

Classes	Precision	Recall	**F-measure(%)**
Person	87.2	85	86
Location	86	84	84.8
Organization	85.1	86	85.5
Miscellaneous	88	87	87.4

The overall performance of NER for Assamese languages using different approaches is shown in Fig 4.3.

Figure 4.2: Hybrid NER Architecture

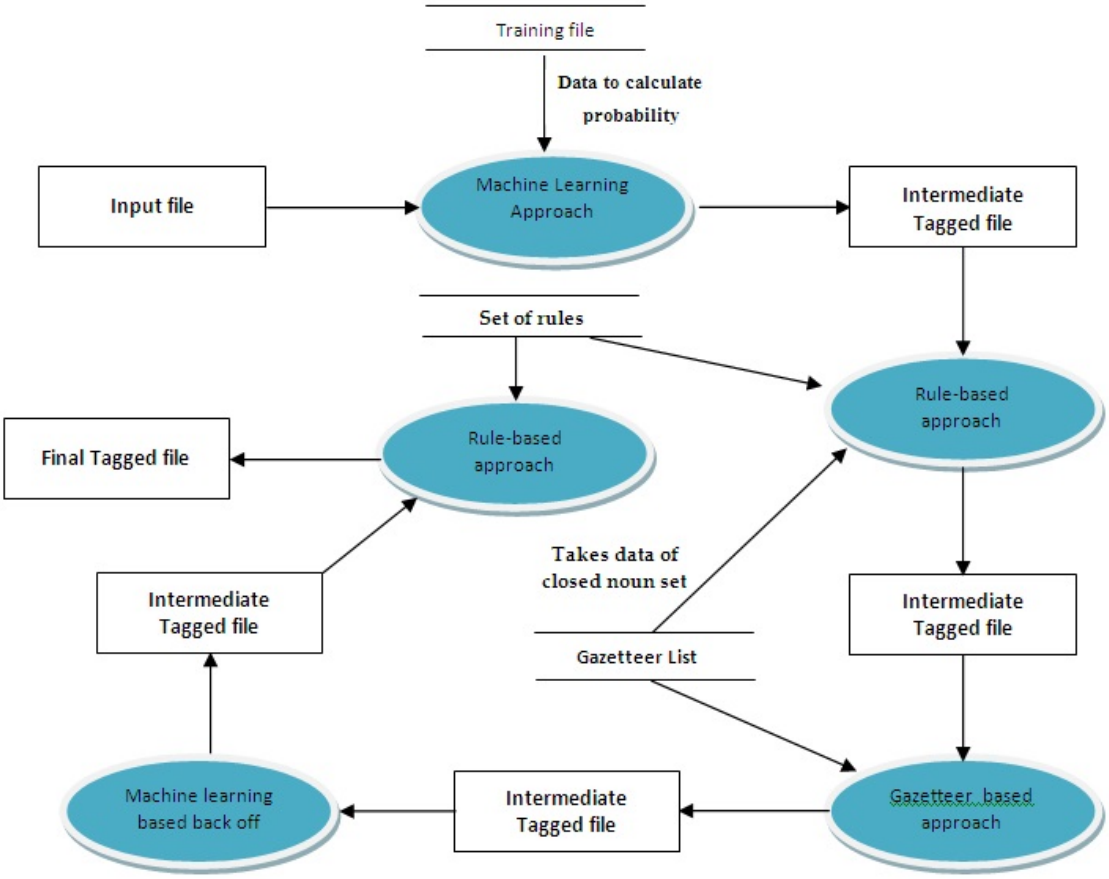


Figure 4.3: Comparison of NER for Assamese using different approaches.

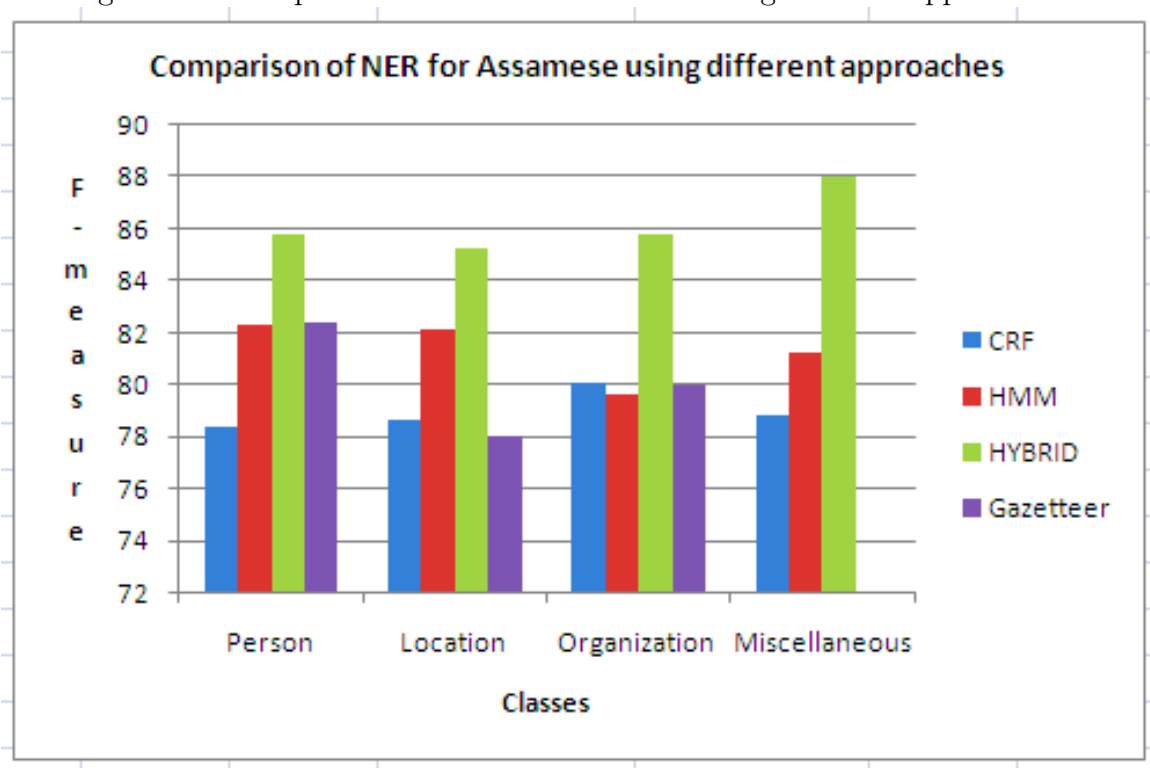


Table 4.19: NER results for Set 3 using Hybrid Approach

Classes	Precision	Recall	**F-measure(%)**
Person	86.1	84	85
Location	85	87	85.9
Organization	85.1	86	85.5
Miscellaneous	89.1	87	88

Table 4.20: Average Result for Hybrid Approach

Classes	**F-measure(%)**
Person	85.8
Location	85.25
Organization	85.8
Miscellaneous	88.06

4.5.1 Previous work On NER using Hybrid Approach

Some of the work found in NER in Indian languages using hybrid approach are briefly described below. Bajwa and Kaur[9] proposed NER for Punjabi using a hybrid approach in which rules are used with HMM. Saha et al.[105] describ a hybrid system that apply the Maximum Entropy Model, language-specific rules and a gazetteer list for several Indian languages. Jahan et al.[55] present a combination of HMM and gazetteer methods for a tourism corpus. Srivastava et al.[126] discuss NER for Hindi using the CRF, HMM and rule-based approaches. Amarappa and Sathyanarayana[5] use the HMM and rule-based approaches for Kannada. Jimmy and Kaur[68] propose a

hybrid approach in Manipuri, combining CRF and rule-based approaches. The accuracies obtained by different authors for different languages using hybrid approaches are shown in Table 4.21. Some more work on Bengali and Hindi using Hybrid approach is also shown in Table 4.22. We see that accuracy varies across languages. Differences in the datasets, sizes of the training data and the use of POS, morphological information, language-specific rules, and gazetteers are the main reasons for the low performance of the systems.

Table 4.21: Different work on NER using Hybrid approach

Reference	**Language**	**Approach**	**F-measure(%)**
Bajwa and Kaur[9]	Punjabi	HMM+Rule-based	74.56
Amarappa and Sathyanarayana[5]	Kannada	HMM+Rule-based	94.85
Srivastava et al.[126]	Hindi	CRF+ME+Rule-based	82.95
Saha et al.[105]	Hindi	ME+Rule-based	65.13
	Bengali	ME+Rule-based	65.96
	Oriya	ME+Rule-based	44.65
	Telugu	ME+Rule-based	18.74
	Urdu	ME+Rule-based	35.47
Jimmy and Kaur[68]	Manipuri	CRF+Rule-based	93.3

Comparison of different work on NER using Hybrid approach in Bengali and Hindi is shown in the Table 4.22 .

We also intend to implement some of the hybrid methods used by different authors on our Assamese data. In Jahan et al.[55], the author use gazetteer and HMM method as a hybrid approach. Firstly they perform gazetteer method

Table 4.22: Different work on NER in Bengali and Hindi using Hybrid approach

Language	**Reference**	**Approach**	**F-measure(%)**
Bengali	Saha et al.[105]	ME+Rule-based	65.96
Hindi	Srivastava et al.[126]	CRF+ME+Rule-based	82.95
	Saha et al.[105]	ME+Rule-based	65.13
	Chopra et al.[23]	HMM+Rule-based	94.61
	Jahan et al.[55]	HMM+Gazetteer	98.37
	Kaur and Kaur[62]	Rule-based+List-look-up	96
	Singh et al.[123]	Hybrid Morphological Analyzer	75-85

on 100 sentences in which the accuracy came out to be 40.13%. Further when HMM is applied on this sentences, the accuracy increases to 93.8%. They then apply the hybrid approach on 40 sentences in which first gazetteer method is used and after that in the remaining tags the HMM approach is used in which the accuracy increases to 98.37%. We have also used the same approach as used by Jahan et al.[55] over the Assamese dataset. A corpus of 100K wordforms is used. When we perform the gazetteer approach our results came to be 75% to 83% which is shown in the Table 4.23. When HMM alone is used on the same dataset, the accuracy came to be 79%-83% as shown in Table 4.24. Now combining both the approaches the result is shown in Table 4.25.

Similarly in Bajwa and Kaur[9] use both rule-based and HMM as an hybrid approach. In the first phase which constitutes of HMM, the accuracy comes out to be 48.27% and when the output of HMM is further optimized with handcrafted rules the accuracy comes to be 74.56%. When we perform the

Table 4.23: Results obtained using gazetteer list

Classes	**F-measure(%))**
Person	82.4
Location	78
Organization	80

Table 4.24: Average HMM Results

Classes	**F-measure(%)**
Person	82.33
Location	82.13
Organization	79.6
Miscellaneous	81.25

Table 4.25: Average Hybrid Results

Classes	**F-measure(%)**
Person	83.4
Location	82.1
Organization	80.4
Miscellaneous	83.3

same appraoch as used by Bajwa and Kaur[9] on our Assamese data, the accuracy using Hybrid approach is shown in the Table 4.26.

After obtaining improved results in Assamese, we extend our work on NER

Table 4.26: Results for Assamese using hybrid approach

Classes	**F-measure(%)**
Person	78.3
Location	82.1
Organization	81.4
Miscellaneous	83.2

to two other Indo-Aryan languages, namely Bengali and Hindi. Bengali is the national language in Bangladesh and second most spoken language in India. It is the seventh most spoken language in the world by total number of native speakers and the eleventh most spoken language language by the total number of speakers. Although the Bengali script is similar to the Assamese script, there are some differences in the scripts such as the Assamese consonant ৰ (ra) is distinct, and Assamese ৱ্ (wabo) is not found in Bengali.

Hindi is spoken by 294.4 million as the first language as per 2001 data, and 366 million worldwide. It is spoken as a native language in northern India. Hindi is written using Devanagiri script. Like Assamese, Bengali and Hindi also lack the concept of capitalization which makes it difficult to identify proper nouns. Names in Hindi and Bengali are also ambiguous and there are variations in the spellings of proper nouns and both these languages lack labeled data. We have manually prepared a training file of 3K word forms for both Bengali and Hindi. The hybrid NER system is trained and tested on these datasets to show the effectiveness of the language independent approaches. Both Bengali and Hindi are tagged with four classes of NE namely person, location, organization and miscellaneous. The result obtained using our Hybrid approach on both

Bengali and Hindi datasets are shown in Table 4.27 and Table 4.28.

Table 4.27: NER Results fo Bengali Dataset using an hybrid approach

Classes	Precision	Recall	**F-measure(%)**
Person	62.3	79	69.6
Location	71	80	75.2
Organization	63.5	75	68.7
Miscellaneous	82	88	84.8

Table 4.28: NER Results for Hindi Dataset using an hybrid approach

Classes	Precision	Recall	**F-measure(%)**
Person	65	72	68.3
Location	75.3	79	77.1
Organization	68	72.1	69.9
Miscellaneous	79.3	80.4	79.8

4.5.2 Results Discussions

We have tested our hybrid approach on two Indian languages namely Bengali and Hindi. Although not much work can be found in these two languages using hybrid approach yet we have listed some work which is shown in Table 4.22. We see that the accuracies of the system for both Bengali and Hindi usng our hybrid approach is lower compared to Assamese, which is due to the smaller amount of the training data. We believe that the performance of the

system will increase on increasing the size of the corpus. We have seen that gazetteer method is more effective for smaller data set whereas HMM perform better for larger dataset. Thus when we perform experiment using gazetteer first and then applying HMM for a small set of data, the performance of NER is much better whereas for a larger set of data when we apply HMM method first and then gazetteer the results show improvement. This is due to the fact that larger the dataset higher the ambiguity which is difficult to resolve using the gazetteer method.

4.6 Conclusion

This chapter discusses two ML approaches, namely CRF and HMM. We have used various language dependent and independent features when implementing the approaches. Both CRFs and HMM based NER systems perform well, but we encountered problems, which are overcome to some extent using a hybrid approach. We can conclude that statistical approach works well for a larger set of data compared to a small set of data whereas rule-based and gazetteer works well for smaller set of data. In our Hybrid approach, we have implemented different approaches in a sequential manner based on careful analysis i.e., which approach is to be implemented first, and then applied the other method on the left out data. This results in higher percentage in hybrid approach. Thus we can say that from our experiments compared to a single statistical approach, a combined approach gives much better results. We also implement our hybrid approach on two other Indian languages namely Bengali and Hindi and came out with an acceptable results.