Chapter 1

Introduction

1.1 Introduction

By Natural Language Processing(NLP), we mean the computational techniques that process written or spoken human languages. In particular, NLP is the branch of computer science focused on developing systems that enable computers to communicate with people using everyday language. It is also called Computational Linguistics. The Internet-focused world we live in generates a large amount of information every-day and access to such a system has changed the way we live and work. The abundant data is useful only if suitable techniques are available to process the data and extract knowledge from it. This is termed Information Extraction (IE). IE plays an important role in NLP in transforming unstructured textual data into structured data that can be understood by machines. An important step in this regard came into existence at the Message Understanding Conference Grishman and Sundheim[48], whose main goal was to identify entities which can be considered names from a set of documents and classify them into predefined categories. This

process is called Named Entity Recognition (NER). A lot of work has been done in NER in English and other European languages and only a limited work can be found in Indian languages like Bengali, Hindi, Tamil, Telugu, Kannada, Urdu, Nepali etc. But till now no work has been reported in Assamese which is a resource poor Indo-Aryan language. We are the first one to undertake the work.

1.1.1 Definition of Named Entity Recognition

A Named Entity (NE) is an element in text that refers to the name of a particular item such as a person, organization, or location. Recognition and tagging of NEs in text is an essential component of NLP tasks such as IE, Question Answering (QA), and Automatic Summarization (AS).

In the MUCs, Ralph[98], clarified that it is necessary to first identify certain classes of NE in order to extract information from a given document. Later, the conference established the NER tasks Chinchor et al.[22]. Systems were asked to identify names, dates, times, and numerical information. NER can be defined as a two-stage problem: identification of proper nouns; and the further classification of these proper nouns into a set of classes such as person names, location names (e.g., cities and co untries), organization names (e.g., companies, government organizations, and committees), and miscellaneous names (e.g., date, time, number, percentage, monetary expressions, number expressions, and measurement expressions).

A few conventions for tagging NEs were established in the MUC Conferences Chinchor et al.[22]. These include ENAMEX for names (organizations, persons, locations, NUMEX for numerical entities (monetary, percentages), and TIMEX tags for temporal entities (time, date, year). For example consider the sentence:

This Memorandum of Understanding (MOU) between Computer Society

of India and DOEACC Society, New Delhi was signed on 29th July 2010 by Dr. S. Birendra Singh, Executive Director, DOEACC Society [1].

Using an XML format, it can be marked up as follows:

This Memorandum of Understanding (MOU) between <ENAMEX TYPE="ORGANIZATION">Computer Society of India</ENAMEX> and <ENAMEX TYPE="ORGANIZATION">DOEACC Society,</ENAMEX> <ENAMEX TYPE="LOCATION">New Delhi</ENAMEX>was signed on <TIMEX TYPE="DATE">29th July 2010 </TIMEX>by<ENAMEX TYPE="PERSON">Dr. S. Birendra Singh</ENAMEX>Executive Director,<ENAMEX TYPE="ORGANIZATION">DOEACC Society</ENAMEX>.

Here, the markups show the named entities in the document.

1.1.2 Application of Named Entity Recognition

NER has been applied in many applications such as IE, QA, Event Extraction and Web Mining.

1. *Information Extraction*:- IE is automatic extraction of structured information from an unstructured document. NER can be defined as an IE task which is concerned with the identification and classification of proper nouns in a predefined set of categories. Additionally, it includes the extraction of descriptive information from the text about the entities. Examples may include the extraction of person title, designation, date of birth, and other attributes of a person.

2. *Question Answering*:- NER is widely used in the field of QA, which is the process of automatically finding an answer to a question by searching through a collection of documents. Molla et al.[83] discuss the use of NER in QA and report that maintaining a high recall is an important task in QA in their AnswerFinder Project (AFNER).

   Besides these, NER can also be applied in co-reference resolution, Web mining, molecular biology, bioinformatics, and medicine. Maynard et al.[77] designed a system for scientific mail text and religious text and Minkov et al.[82] created an NER system for email documents.

3. *Event Extraction*:- NER also plays an important role in event extraction which involves the detection of entities and relationships between them. Extracting events requires the ability to recognize the NEs, e.g., conferences are usually comprised of topics, dates, venues, etc.

1.2 Problems in Named Entity Recognition

NER was first introduced as part of MUC-6 in 1995. Generally, NEs have different structures across different languages. Names can overlap with other names and other words. Human language being polysemous, proper identification and classification of names is important task as there are ambiguities among different classes. Thus, the goal of NER is to recognize the NEs and then resolve the ambiguities in the names. The task of NER in general faces the challenge of ambiguity. Consider the sentence:

Rose rose to put rose roes on her rows of roses.[1].

In this sentence *rose* can either be a person name or a common noun, making it difficult for the computer to resolve the ambiguity between the two. A program is forced to use domain or linguistic knowledge in the form of rules, such as the name of a person (which is a proper noun) usually begins with a capital letter, in order to resolve the issues. Thus, the correct NE annotation of the above sentence can be:

<ENAMEX TYPE="PERSON">Rose</ENAMEX> rose to put rose roes on her rows of roses.

Domain or linguistic knowledge can come in other ways also, e.g., statistics. Another form of ambiguity is that frequently there are overlaps among classes of NEs. Ambiguity is one of the main challenges in NER. The different types of ambiguity that occurs in NER are as follows:

- *Person vs. location*:- In English, a word such as *Washington* or *Cleveland* can be the name of a person or a location. Similarly, in Indian English (or in an Indian language when written in the appropriate script), words such as *Kashi* can be a person name as well as a location name.
- Words or word sequences also exist such as *Thinking Machines* (a company), or *Gates* (a person), that can occur in contexts where they do not refer to NEs.
- *Common noun vs. proper noun*:- Common nouns sometimes occur as a person name such as *Suraj* which means sun, thus creating ambiguities between common nouns and proper nouns.

[1]http://www.wikipedia.org/wiki/List_of_linguistic_example_sentences

- *Organization vs. person name*:- *Amulya* may be the name of a person as well as that of an organization, creating ambiguity.

- *Nested entities*:- Nested entities such as *New York University*, also create ambiguity because they contain two or more proper nouns.

Such phenomena are abundant in Indian languages as well. These ambiguities in names can be categorized as structural ambiguity and semantic ambiguity. Wacholder et al.[131] describe such ambiguities in detail. A number of additional challenges need to be addressed in South Asian languages such as Assamese, Hindi, Bengali, Telugu, Urdu and Tamil. The key challenges are briefly described as follows. Although our examples are in specific languages, similar phenomena occur in all Indian languages and Assamese in particular.

- *Lack of capitalization*:- Capitalization plays a major role in identifying NEs in English and some other European languages. However, Indian languages do not have the concept of capitalization.

- *Ambiguity*:- In Indian languages, the problem of ambiguity between common nouns and proper nouns is more difficult since names of people are usually dictionary words, unlike Western names. For example, আকাশ [akas] and জোন [zun] mean *sky* and *moon*, respectively, in Assamese, but also can indicate person names. In fact most people's names are dictionary words, used without capitalization.

- *Nested entities*:- Indian languages also face the problem of nested entities. Consider, in Assamese তেজপুৰ বিশ্ববিদ্যালয় [tɛzpuɾ bisʱɒbidyalɒi] [E:[2]Tezpur University]. It creates a problem for NER in the sense that the word তেজপুৰ

[2]E: English meaning

[tɛzpuɾ] [E:Tezpur] refers to a location, whereas বিশ্ববিদ্যালয় [bisʰɒbidyalɒi] [E: University] is a common noun and thus তেজপুৰ বিশ্ববিদ্যালয় [tɛzpuɾ bisʰɒbidyalɒi] [E:Tezpur University] is an organization name. Thus it becomes difficult to retain the proper class.

- *Agglutinative nature*:- Agglutination adds additional features in the root word to produce complex meaning. Consider the following example 1, in Telugu హైదరాబాద్ (E: Hyderabad) refers to a named entity whereas హైదరాబాద్‌లోనుంచి(E:Hyderabadlonunci) is not a named entity as it refers to anyone who is a resident of Hyderabad.

  హైదరాబాద్‌లోనుంచి(Hyderabadlonunci)=హైదరాబాద్ (Hyderabad)+ లో(lo)+ నుంచి(nunci)

  Similarly in Assamese, গুৱাহাটী [guwahati] [E:Guwahati] refers to a location named entity whereas গুৱাহাটীয়া [guwahatiya] [E:Guwahatiya] is not a named entity as it refers to the people who stay in Guwahati.

  In NER we have to identify named entities which may have occurred as compound words with some other word, or with suffixes. This requires finding the base form of the words.

- *Ambiguity in suffixes*:- Indian languages can have a number of postpositions attached to a root word to form a single word. Consider Example 2, in Telugu, guruvAraMwo which means in English (E:up to Wednesday). We can have the word

  Example2:

  గురువారం(TF:guruvaram) +ఒ (TF:wo)= గురువారంఒ (TF:guruvaramwo)

  which means guruvAraMwo (E:up to Wednesday) = guruvAraM (E:Wednesday) + wo (E:up to). This creates a problem for NER in the sense that the suffix (wo) when added to the root word gives

a different meaning when compared to the original word. In Assamese the word মনিপুৰ [mɒnipuɾ] [E:Manipur] is a place name, but when the suffixী [ee] is attached, it gives a different meaning compared to the original one which means the people of Manipur. Sometimes separating two compound words or suffixes is non-trivial.

- *Resource constraints*:- NER approaches are either rule based or machine learning(ML)-based. In either case, a good-sized corpus of the language under consideration is required. Such corpora of significant size are still lacking for most Indian languages. Basic resources such as parts of speech (POS) taggers, or good morphological analyzers, and name lists, for most Indian languages do not exist or are in research stages, whereas a number of resources are available in English.

- *Foreign Words*:- Often names of entities are language specific, for example, State Bank of India, and when such an entity is referred to in another language text, it has either to be translated to that language or transliterated. Transliteration is the process of writing a source language expression in a target language script based on phonetic similarity. For example, *tumar naam ki?* (meaning: what is your name) and *ami bhalo achi* (meaning: I am fine) are Roman transliterations of an Assamese and a Bengali sentence, respectively. Technical terms and NEs make up the bulk of Out of Vocubulary (OOV) words. NE phrases are the most difficult to translate because new phrases are continuously coined, they are domain specific, and are usually not found in bilingual dictionaries. Simple literary translations of some NEs do not make sense, for example, Air India, All India Radio, etc. Transliteration is important for NEs, even when the word could be translated. Moreover, an increasing number of OOV words in a text originally are from another language, either

in the foreign script or transliterated to the host language script. Person names, location names, organization names, etc. must be available to users of vernacular languages in their own scripts. Transliteration is required for NE for language pairs that use different writing systems. For example, in Assamese the name *Ram* is written as রাম.

1.3 Approaches to Named Entity Recognition

Techniques for NER can be classified in three methods:

1. Rule-based approaches;
2. Machine Learning approaches; and
3. Hybrid approaches.

Rule-based NER focuses on the extraction of names using human-made rules. A rule-based system requires a human expert to define rules in which the person needs to be a domain expert and have good programming skills. This method is easier to develop and interpret than statistical methods. In general, the rule-based approach consists of a set of patterns using grammatical, syntactic and orthographic features. This approach lacks portability and robustness. One needs a significant number of rules to maintain optimal performance, resulting in high maintenance cost. There are several rule-based NER systems for English providing 88%-92% F-measure [Wakao et al.[132], Ralph et al.[98]]. Appelt et al.[7] proposed a model name FASTUS, which is a name identification system. LASIE by Kaufmann et al.[60] and LASIE II Humphreys et al.[59] used the concept of a look-up dictionary and grammatical rules to identify the NEs. As noted by Nadeau[86], rule-based approaches were

primarily adopted by early entity recognition systems. The main attraction of the ML approach is that it is trainable and can be adapted to different domains. The maintenance cost can also be less than that of the rule-based approach. In developing a rule-based system, manual effort is required to carefully create the rules. On the other hand, in machine-learning approach manual effort is required to create the training data– corpus in case of NLP. With access to electronic text available, corpus creation can be less taxing than rule-base creation, particularly when unsupervised or semi-supervised ML approach is used.

The ML approach identifies proper names by employing statistical models that classify them. ML models can be broadly classified into three types:

1. Supervised;
2. Unsupervised; and
3. Semi-supervised.

1. *Supervised*: In supervised learning, the training data include both the input and the output. In this approach, the construction of proper training, validation and test sets is crucial. This method is usually fast and accurate. As the program is taught with the right examples, it is "supervised". A large amount of training data is required for good performance of this model. Several supervised models used in NER are: Hidden markov Model(HMM) [Bikel et al.[74],Miller et al.[81],Yu et al.[135]]; Conditional Random Field(CRF) Lafferty et al.[69]; Support Vector Machine(SVM) Cortes and Vapnik[25]; and Maximum Entropy(ME) Borthwick[6]. In addition, a variant of Brill's transformation-based rules Brill[19] has been applied to the problem Aberdeen et al.[2]. HMM is widely used in NER due to the efficiency of the Viterbi

algorithm Viterbi[130] used to discover the most likely NE class state sequence.

Hidden Markov Model (HMM)

HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved states. An HMM can be thought of as a simple dynamic Bayesian network. In this approach the state is not directly visible, but, the output depends on the state and is visible. Each state has a probability distribution over the possible output tokens. The sequence of tokens generated by an HMM gives information about the sequence of states. The word "Hidden" refers to the state sequence through which the model passes. Instead of single independent decisions, the model considers a sequence of decisions.

An HMM is defined as follows: given a sequence of word (W), we need to find the most likely sequence of name-classes (NC)Bikel et al.[74],i.e.,

$$max\,\Pr(NC|W), \tag{1.1}$$

by using Bayes theorem:

$$\Pr(NC|W) = \frac{\Pr(W, NC)}{\Pr(W)}. \tag{1.2}$$

Now, as Pr(W), the unconditioned probability of any word sequence, is constant, hence maximize the numerator, i.e., the joint probability of the word and name-class sequence.

In the field of NER, HMM is used to compute the likelihood of words occurring within a given category of NE. Zhou and Su[136] used HMMs for NER in English. Biswas et al.[17] used HMM for the Odiya language. Biswas et al.[112]

also described the development of a two-stage hybrid NER system for several Indian languages using MEs and HMMs. Ekbal and Bandyopadhyay[32] discussed both Bengali and Hindi as case studies using HMMs whereas Kumar and Kiran [90]used HMM for a Hybrid NER system for Hindi, Bengali, Urdu, Tamil, and Telugu, and Pandian et al.[92] for Tamil. The following are the assumptions associated with an HMM:

- Each state depends on its immediate predecessor;
- Each observation value depends on the current state; and
- One needs to enumerate all observations.

There are three characteristic problems associated with HMMs.

- Given the parameters of the model, to compute the probability of a particular output sequence requires summation over all possible state sequences. This can be done efficiently using the forward or the backward algorithm Rabiner and Juang[102].
- Given the parameters of the model and a particular output sequence, to find the state sequence that is most likely to have generated the output sequence requires finding a state sequence with the highest probability. This can be solved efficiently using the Viterbi algorithm Viterbi[130].
- Given an output sequence or a set of such sequences, and the topology one needs to find the parameters that define an HMM, One can find the maximum likelihood estimates of the parameters of the HMM efficiently using the expectation maximization algorithm Rabiner and Juang[102].

Conditional Random Field (CRF)

CRFs Lafferty et al.[69] are a type of discriminative probabilistic models used for labelling and segmenting sequential data such as natural language text

or biological sequences. CRFs represent an undirected graphical model that defines 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 values on other designated input nodes. The conditional probability of a state sequence $S = (s1, s2..sT)$ given an observation sequence $O = (o1, o2, o3...ot)$ is calculated as:

$$P(s|o) = \frac{1}{Z_o} \exp(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(S_{t-1}, S_t, o, t)) \tag{1.3}$$

where Z_o is a normalization factor overall state sequence,

$$Z_o = \sum \exp(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(S_{t-1}, S_t, o, t)) \tag{1.4}$$

and $f_k(S_{t-1}, S_t, o, t)$ is a feature function whose weight λ_k is to be learned via training.

Comparison with the other models

- CRFs avoid the problem of label bias Lafferty et al.[69], a weakness of the Maximum Entropy Markov Model (MEMM) McCallum et al.[78] and other conditional Markov Models based on directed graphs.
- CRFs outperformed both MEMM and HMM in a number of real world sequence labelling tasks [Pinto et al.[93],Sha and Pereira[115]].

When applying CRFs to the NER problem, an observation sequence is a token of a sentence or document of text and the state sequence is its corresponding label sequence. CRFs have been used for several Indian languages in the NER task. For example Ekbal et al.[42] and Gali et al.[46] used CRF for

Hindi, Bengali, Tamil, Telugu, Odiya, and Urdu. Ekbal et al.[41] used them for Bengali; Krishnarao et al.[66] for Hindi; Shishtla et al.[120] for Telugu and Hindi; Vijaykrishna and Sobha[96] for Tamil; Shishtla et al.[119], and Srikanth and Murthy[125] for Telugu; and Mukund and Srihari[85] for Urdu.

Support Vector Machines (SVM)

SVMs represent a relatively new ML approach that analyzes data and recognizes patterns for classification and regression analysis. The original SVM algorithm was proposed by Vapnik Cortes and Vapnik[25].

In the field of NLP, SVMs have been applied to text categorization by Taira and Haruna[128] and many other problems and are known to produce excellent results. An SVM performs classification by constructing an n-dimensional hyperplane that separates the data optimally into two categories for the binary case. A set of features that describe one example is called a vector. Thus, the goal of an SVM is to find the optimal hyperplane that separates clusters of vectors in such a way that examples in one category of the target variable are on one side of the plane. The vectors near the hyperplane are the support vectors. Suppose we have a set of training data for a two-class problem:

$(x_1, y_1)...(x_n, y_n)$, where x_i is a feature vector of the i^{th} sample in the training data and $y \in (+1, -1)$ is a Boolean variable that specifies class membership. +1 means membership in a designated class and -1 means non-membership.

Given $x \in X$, find a suitable $y \in Y$ i.e, to learn a classifier $y = f(x, \alpha)$, where α are the parameters of the function. For example, if we choose a model from the set of hyperplanes in R^n, it can be written as:

$$f(x, w, b) = sign(w.x + b) \tag{1.5}$$

where x is the example to be classified, w is a weight vector, and b is the bias which stands for the distance of the hyperplane to the origin. In NER, SVM takes an input of a set of training examples (given as binary-valued feature vectors) and finds a classification function that maps them to a class.
The SVM has certain advantages over other models such as HMMs and ME.

- It is an attractive method due to high generalization ability and its ability to handle high-dimensional input data.
- SVMs deliver a unique solution, since the optimality problem is convex. This is an advantage compared to other techniques, which have multiple solutions associated with local minima and for this reason may not be robust over different samples.

However, SVMs suffer from slow training with non-linear kernels and with large input data size. In Ekbal and Bandyopadhyay[36],Ekbal and Bandyopadhyay[35]], the authors used SVMs for Bengali, whereas in Ekbal and Bandyopadhyay[40] they used the same approach for Hindi. Krishnarao et al.[66] have used this approach for Hindi as well.

Maximum Entropy (ME)

The principle of ME states that subject to known constraints, the probability distribution which best represents the current state of knowledge is the one with largest entropy. The ME method aims at providing a model with the least bias possible Berger et al.[16]. The ME framework estimates probabilities based on the principle of making as few assumptions as possible other than the constraints imposed. Such constraints are derived from training data,

expressing relationships among features and outcomes.

$$P(o|h) = \frac{1}{z(h)} \prod_{j=1}^{k} \propto_j f_j(h, o) \tag{1.6}$$

where o refers to the outcome, h the history (or context) and $z(h)$ is a normalization function. In addition, each feature function $f_j(h, o)$ is a binary function. The parameters $\propto_j$ are estimated by a procedure called Generalized Iterative Scaling (GIS) Darroch and Ratcliff[26]. This is an iterative method that improves the estimation of the parameter at each iteration. When applying ME in NER, the future is the possible output of the model, the history is the words in the tokenized training corpus and the features include the history to assign probability distribution to the future using some features. In NER, Biswas et al.[17] have used ME for Odiya; Raju et al.[97] for Telugu; Saha et al.[105] for Hindi; and Hasanuzzaman et al.[52] for Bengali and Hindi.

2. *Unsupervised*:- Unsupervised learning[24] refers to techniques that find patterns in unlabeled data, or data that lacks a defined response measure. The goal for unsupervised learning is to model the underlying structure or distribution in the data in order to learn more about the data. Unsupervised learning studies how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. Model builds a representation from the data. It is called unsupervised learning because unlike supervised learning there is no correct answers and there is no teacher. The algorithms discover the interesting structures in the data. Unsupervised learning problems can be further grouped into clustering and association problems.

    - Clustering: A clustering problem is where we want to discover the inherent groupings in the data, such as grouping customers by purchasing

behavior. In this type of learning, the goal is not to maximize a utility function, but simply to find similarities in the training data. The assumption is often that the clusters discovered will match reasonably well with an intuitive classification. For instance, clustering individuals based on demographics might result in a clustering of the wealthy in one group and the poor in another. For example, one can try to gather named entities from clustered groups based on the similarity of context.

- Association: An association rule learning problem is where we want to discover rules that describe large portions of data, such as people that buy X also tend to buy Y.

Some popular examples of unsupervised learning algorithms are: k-means for clustering problems and Apriori algorithm for association rule learning problems. There are other unsupervised methods too. Basically, the techniques rely on lexical resources e.g., WordNet, on lexical patterns and on statistics computed on a large unannotated corpus. Here are some examples. Alfonseca and Manandhar[4] study the problem of labelling an input word with an appropriate NE where the NEs are taken from the WordNet. Sekine et al.[113] used an observation where named entities often appear synchronously in several documents whereas common nouns do not. A strong correlation is found between being a named entity and appearing punctually (in time) and simultaneously in multiple news sources which allows identifying rare named entities in an unsupervised manner and can be useful in combination with other NER methods. Collins and Singer[24] discuss an unsupervised model for NE classification using unlabeled data. It is possible to learn larger and more complex models with unsupervised learning than with supervised learning. This approach is portable to different domains or languages unlike the rule-based approach.

3. *Semi-supervised*:- Problems which have a large amount of input data (X) and only some of the data is labeled (Y) are called semi-supervised learning[20] problems. As the name suggests, semi-supervised learning is somewhere between unsupervised and supervised learning. The semi-supervised model makes use of both labeled and unlabeled data which results in high accuracy. Here, expertise is required to obtain labeled data. The cost of labeling the data is high.It has been found that unlabelled data when used in conjunction with a small amount of labelled data results in improvement in accuracy. Semi-supervised learning may refer to either transductive learning or inductive learning where transductive learning infers the correct label for the given unlabelled data whereas inductive learning infers the correct mapping from X to Y. One can use unsupervised learning techniques to discover and learn the structure in the input variables and it can also be used to make best guess predictions for the unlabeled data, feed that data back into the supervised learning algorithm as training data and use the model to make predictions on new unseen data. Semi-supervised learning algorithm make use of at least one of the following assumptions:

    - Smoothness assumption: Here the points which are close to each other are more likely to share a label.
    - Cluster assumption: Here the data tend to form discrete clusters and points in the same clusters are more likely to share a label.
    - Manifold assumptions: Here the data lie on a low dimensional manifold embedded in a higher dimensional space. .

    A good example is a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled.

    Many real world machine learning problems fall into this area. This is because

it can be expensive or time consuming to label data as it may require access to domain experts. Whereas unlabeled data is cheap and easy to collect and store. Bootstrapping is a popular approach for this method. A work in NER using semi-supervised method can be found in Liao and Veeramachaneni[72] where they have used CRF. The algorithm is based on exploiting evidence that is independent from the features used for a classifier, which provides high-precision labels to unlabeled data. Finally the system achieves an average improvement of 12 in recall and 4 in precision compared to the supervised algorithm. Their algorithm achieves high accuracy when the training and test sets are from different domains. Algorithms such as co-training Blum and Mitchell[18], Collins and Singer[79] and Nadeau and Sekine[87] make assumptions about the data that permit such an approach.

1.4 Conclusion

This chapter gives a brief introduction to NER and the different issues that arise in NER. There are different approaches or methodologies to NER, which are introduced in this chapter. The objective of this dissertation is to develop a computational system that can perform NER in the text of the Assamese language, which is a resource-poor Indo-Aryan language. Assamese is a highly inflectional and morphologically rich language. It is one of the national languages of India and is spoken by over 30 million people in North-east India. Broadly, our thesis addresses different issues related to NER in general and Assamese NER in particular. We implement different automated approaches together with handcrafted rules, and finally suggest a suitable approach for Assamese NER. Till now to the best of our knowledge no work on Assamese NER has been reported. Our's is the first such

work although a lot of work can be seen in other Indian languages. We do not claim that our approach is suitable for all Indian languages. Different languages have characteristics that require individual research attention.