# Chapter 2

# NER Research in Indian Languages

## 2.1 Introduction

NER has made remarkable progress in European languages [Molla et al.[83],Florian et al.[43],Yang et al.[133]]but only limited work is found in Indian languages. We have come across reports on NER work in seven Indian languages, namely Bengali, Hindi, Tamil, Telugu, Odiya, Urdu, and Punjabi but no work can be found in Assamese till now. We initiated the work. Each of these languages is an official language of the Republic of India. The numbers of speakers has been obtained from the English Wikipedia[1] for these languages. Bengali, Hindi, and Odiya belong to the Indo-Iranian branch of Indo-European languages. Tamil and Telugu are Dravidian languages. We discuss the work of NER in these languages, the datasets and the tagsets along with the approaches used by the different researchers, and the accuracies obtained. To evaluate the performance of NER, different metrics are also

[1]http://www.wikipedia.org/wiki/Languages-by-speakers

used which is briefly described in this chapter.

## 2.2 NER work in Indian languages

Bengali, an eastern Indo-Aryan language, is the sixth-most popular language in the world and the second-most popular in India. It is the national language of Bangladesh. It is spoken by 220 million people as the native language and has about 300 million total speakers worldwide.

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. It is the fourth-most widely spoken language in the world in combination with Urdu.

Tamil is a Dravidian language that has official status in the Indian state of Tamil Nadu and in the Indian Union Territory of Puducherry (formerly Pondichery). It was the first Indian language to be declared a classical language by the Government of India in 2004. It is spoken by between 61 and 78 million people worldwide.

Telugu is a Dravidian language. It has the third-largest number of native speakers in India. It is spoken as a first language by 74 million people and by 5 million people as a second language in India, primarily in the state of Andhra Pradesh (2001 Census). Telugu is one of the 22 official languages and 14 regional languages of India. It is the official language of the state of Andhra Pradesh and has also been conferred the status of a classical language by the Government of India.

Odiya is mainly spoken in the Indian state of Odisha. Odiya, also known as Odia, has nearly 40 million native speakers in the world.

Urdu is the national language of Pakistan, and one of its two official languages spoken by a total of 104 million people in the world. It is also an official language of India. It is very similar to Hindi. Urdu is written in Nastaliq calligraphy style of

the Perso-Arabic script whereas standard Hindi is written in Devanagari. However, both have large numbers of Persian, Sanskrit, and Arabic words. Urdu belongs to the Indo-Iranian branch of the Indo-European family of languages.

Punjabi is an Indo-Aryan official language in India. It is the first official language of Punjab (India) and the Union Territory of Chandigarh and the second official language of Haryana, Himachal Pradesh, and Delhi. In Pakistan, Punjabi is the most widely spoken language in the country and is the provincial language of Punjab (Pakistan), the most populous province of Pakistan. According to the 2005 estimates, there are 88 million native speakers of the Punjabi language, which makes it the twelfth-most widely spoken language in the world.

### 2.2.1 NER in Bengali

The most extensive amount of NER research in Indian languages has been done in Bengali. We discuss Bengali NER work below by categorizing individual papers according to the technical approach used.

*Shallow-parser Approach*: Ekbal and Bandyopadhyay [34] discussed a semi-supervised learning system based on pattern-directed shallow parsing to identify NEs. The performance was compared for two systems, one using lexical contextual patterns,[2]and the other using linguistic features along with the same set of lexical contextual patterns[3]. They concluded that the use of linguistic knowledge yields a high F-measure of 75.40%, 72.30%, 71.37% and 70.13% for person, location, organization, and miscellaneous names respectively.

*HMM-Approach*: [32] reported an HMM-based NER system, the system developed initially for Bengali using a tagged Bengali news corpus from an online Bengali

[2]NER system without linguistic features

[3]NER system with linguistic features

newspaper. It was then trained with a training corpus of 150K wordforms tagged with an HMM-based POS tagger Ekbal and Bandyopadhyay[33] that used 26 POS tags[4]. This POS-tagged training set was further tagged with 16 NE tags and one non-NE tag. The 16 NE tags were later replaced by four NE tags. Tenfold cross-validation yielded average Recall, Precision, and F-measure values of 90.2%, 79.48%, and 84.5%, respectively. The HMM-based NER system was later trained and tested with Hindi data to show the effectiveness of the language-independent features used by the approach. The results showed average Recall, Precision, and F-measure values of 82.5%, 74.6%, and 78.35%, respectively. The possible reason for the poor performance of the system for Hindi might be the smaller quantum of training data compared to Bengali. Moreover, the Hindi version lacked the ability to handle unknown words and there was no list of suffixes or a lexicon in the Hindi system which Bengali had.

*SVM-Approach*: Ekbal+08[35] studied the use of SVM using contextual features and a several other features such as the previous and following words of a particular word, prefixes, suffixes, POS tags, and lengths of the words and had access to gazetteer lists of NEs. Out of 34 million wordforms, a set of 150K wordforms is tagged manually with 17 tags using the BIE format. Around 20K NE-tagged words were selected as the development set and the remaining 130K wordforms were used as the training set. A tenfold cross-validation test showed the effectiveness of the system with overall average Recall, Precision, and F-measure values of 94.3%, 89.4%, and 91.8% respectively.

The same approach was later used with Hindi Ekbal and Bandyopadhyay[40] and was tagged with 12 NE classes as used in Ekbal and Bandyopadhyay[36]. The system used different items of contextual information of the words along with a variety of orthographic word-level features that were useful for predicting the NE classes. An

[4] `http://www.shiva.iiit.ac.in`

unsupervised algorithm was used to generate the lexical context patterns from an unlabeled corpus of 10 million wordforms and the NER system was tested with gold-standard test sets of 35K and 60K tokens for Bengali and Hindi, respectively. Recall, Precision, and F-measure were 88.61%, 80.12%, and 84.15% respectively for Bengali; and 80.23%, 74.34%, and 77.17% respectively for Hindi. The performance of this system was compared with that of an HMM-based system [32] and it was found that SVM was more efficient than HMM because SVM can effectively handle the diverse and overlapping features of Indian languages which are highly inflectional. However, feature selection plays a crucial role in the SVM framework.

*CRF-Approach*: Ekbal et al.[41] used statistical CRFs to identify and classify NEs into four classes. The system used features as discussed by Ekbal and Bandyopadhyay[35]. A partially NE-tagged Bengali news corpus was used to create the training set. This training set contained 150K wordforms, manually annotated using the same tagset used in Ekbal and Bandyopadhyay[35]. A tenfold cross-validation test showed overall average Recall, Precision, and F-measure values of 93.8%, 87.8%, and 90.7% respectively. The performance of the CRF approach was comparable to that of the SVM model for Bengali as reported in Ekbal and Bandyopadhyay[35], and was substantially better than the HMM model Ekbal and Bandyopadhyay[32] with 6% higher F-measure system due to its better capability to capture morphogically-rich and overlapping features of the Bengali language. Hasan et al.[51] presented a learning-based NE recognizer that doesnot rely on a manually constructed gazetteer. Their experiment showed that induced affix features and the Wikipedia-related features improve a baseline POS tagger and NE tag and when combined in both the results showed an improvement of 7.5% in F-measure over the baseline NE recognizer. Das and Garain [27] discuss CRF-based NER in Indian languages. Different features such as context words, prefixes, suffixes, POS, first, and last words are used, and for English the concept of capitalization is used. The

system achieves an F-measure of 88% for English and 69% for Tamil and Telugu respectively, whereas for Bengali and Hindi it had an accuracy of 87% and 79% respectively.

*ME-Approach*: Hasanuzzaman et al.[52] described the development of an NER system in Bengali and Hindi using the ME framework with 12 NE tags. The system used contextual information for the words along with a variety of orthographic word-level features. The average Recall, Precision, and F-measure were 88.01%, 82.63%, and 85.22% respectively for Bengali; and 86.4%, 79.23%, and 82.66% respectively for Hindi. The author used language-independent as well as language-dependent features for both languages. The use of more language-dependent features and a higher number of NEs in Bengali training data than in Hindi resulted in lower comparative accuracy in Hindi.

*Combination of ME, CRF and SVM Approaches*: Ekbal and Bandyopadhyay[37] combined the outputs of several classifiers based on ME, CRF, and SVM. A corpus consisting of 250K wordforms was tagged manually with four NE classes: 30K of the NE-tagged corpus were selected as the development set and the remaining 220K were used as a training set for each of the classifiers. The system used contextual information along with a variety of features as used in Ekbal and Bandyopadhyay[35]. To improve the performance of each of the classifiers, a semi-automatic context pattern induction method was used. Second-best tags and several heuristics were also used. The overall average Recall, Precision, and F-measure values were 90.78%, 87.35%, and 89.03% respectively. This showed an improvement of 11.8% in F-score over the best-performing SVM-based baseline system and an improvement of 15.11% in F-measure over the least-performing ME-based baseline system. In the second paper, combining various NER methods, Ekbal and Bandyopadhyay[38] once again combined the output of ME, CRF, and SVM classifiers. The features used were similar to those used in Ekbal and

Bandyopadhyay[37]. The training, test, and development sets were also comparable to those used in Ekbal and Bandyopadhyay[37]. The overall Recall, Precision, and F-score values were 87.11%, 83.61%, and 85.32% respectively, which show an improvement of 4.66% in F-measure over the best-performing SVM-based system and an improvement of 9.5% in F-measure over the least-performing ME-based system. In Ekbal and Bandyopadhyay[39], the same authors described a voted NER system by using Appropriate Unlabelled Data. This method was also based on supervised classification using ME, SVM, and CRF. The SVMs used two different methods, known as forward parsing and backward parsing. Once again, the models were combined into a final system by a weighted voting technique to obtain overall Recall, Precision, and F-measure values of 93.81%, 92.18%, and 92.98%, respectively.
*Approach using Dictionary, Rules and Statistics, Margin Infused Relaxed Algorithm*: Chaudhuri and Bhattacharya[21] used a three-stage approach, namely, a dictionary-based method, rules, and left-right co-occurrences statistics for identification of NEs. A corpus of 20 million words of the Anandabazar Patrika from 2001 to 2004 was used, out of which nearly 70K words were used for manual tagging. The average Recall, Precision, and F-measures were 85.50%, 94.24%, and 89.51%, respectively. The authors observed that their automatic evaluation system gave almost the same result as manual evaluation. Compared to statistical learning methods, a rule-based system has some limitations: it cannot tackle ambiguous situations. It was also useless for a word not falling under any of the rules generated. Banerjee et al.[14] describe the automatic recognition of NEs using a Margin-infused Relaxed Algorithm. Various language-dependent and language-independent features are used for the Bengali language. IJCNLP-08 NER on South and South-east Asian Languages (NERSSEAL) shared task data is used. These data consist of 12 NE tags and obtained Precision of 91.23%, Recall 87.29%, and F-measure 89.69%.
*Voting based approaches*: [Ekbal

and Bandyopadhyay[37],Ekbal and Bandyopadhyay[39]] gave much better results than classifier combinations Ekbal and Bandyopadhyay[38] due to the fact that the former used the second-best tags of the classifier and applied several heuristics to improve performance. Moreover, the use of a voting scheme further improved the overall performance of any system.

### 2.2.2 NER In Hindi

The NER task for Hindi was first explored by Cucerzan and Yarowsky in their language-independent NER work using morphological and contextual evidence. The work on Hindi NER is discussed in this section.

*ME approach*: Saha et al.[108] described the development of Hindi NER using the ME approach. The features used for the Hindi NER task included orthographic features, affixes, previous words, next words of a particular word, gazetteer lists for NER, parts-of-speech, and morphological features similar to the Bengali NER. The training data used for the task consisted of 234K words, collected from the newspaper *Dainik Jagaran*, which was manually annotated and had 16,482 NEs. Four types of NEs were considered, namely person, location, organization, and date. The authors used a module for the semi-automatic learning of context patterns where the quality of the pattern was determined by Precision and coverage. The system was evaluated using a blind test corpus of 25K words and provided an F-measure of 81.52%. In Saha et al.[107] the authors achieved F-measure of 80.01% using word selection and word clustering-based feature reduction techniques. Saha et al.[106] used the same features to also develop an ME-based system. However, the authors used a two-phase transliteration methodology to make English lists useful in the Hindi NER task. This transliteration approach was also applied to the Bengali NER task. The highest F-measure achieved by an ME-based system for Hindi was 75.89% which

then increased to 81.12% by using the transliteration-based gazetteer list. The F-measure was 69.59% for Bengali. The difference in the accuracy between these two languages was that the training corpus for Bengali was only 68K words whereas in Hindi the corpus contains 243,K words.

*CRF Approach*: Goyal[47] focused on building Hindi NER using CRFs. He used the NLPAI Machine Learning Contest 2007 data for experiments[5]. Features such as contextual (word window, chunk window), statistical (binary features, trigger words), and word internal features (such as prefix and suffix information) were used for three different modules, namely the Named Entity Recognition (NER) module[6], the Named Entity Classes (NEC) module[7] and the Nested Named Entity (NNE) module[8]. The author used BIO[9] model tagging for NEs. This method was evaluated on two different test sets and it attained a maximum F1-measure of 49.2% and nested F1-measure of 50.1% for test set 1 which contains 1091 sentences; maximum F1-measure 44.97% and nested F1-measure 43.70% for test set 2 which contains 744 sentences; and F-measure of 58.85% on the development set. The difference in the numbers could be due to differences in the test and development sets. The author also compared the results on Hindi data with English data of the CONLL shared task of 2003. They trained this system on English data of the CONLL-2003 shared task, considering only contextual features since they give maximum accuracy. They obtained overall F-measures of 84.09% and 75.81% on the test sets. The big difference in F-measure between Hindi and English data was due to the inconsistency of annotated data in the Hindi dataset. Also, the data used in Hindi was from several domains in which many of the terms do not occur in the corpus more than once. It also showed that the accuracy of POS taggers, chunkers, and

[5]http://ltrc.iiit.ac.in/nlpai_contest07/cgi-bin/index.cgi?topic=3

[6]Identifies whether an entity is NE or not

[7]Identifies the type of label associated with each entity

[8]Identifies the nested named entities

[9]Beginning,Internal and Outside

morphological analyzers was not good in Hindi when compared to English. The differences in the domains of the test set and the development set also led to the poor result for Hindi when compared to English.

Gupta and Arora[49] used a CRF model to develop Hindi NER. They used context features (previous and next words of a particular word), a context word list (every language uses some specific patterns which may act as clue words and the list of such type of words was called a context list), and POS tags. The data used for the training of the model were taken from the tourism domain collected from the Web. The collected data were tagged with a POS tagger and the POS tagged data were manually tagged in BIO format for NER. The authors conducted the experiment on a testing dataset of 150 sentences whereas a model was created with 3K sentences. Finally, after adding the NE tag, the F-measures of the results for person were 66.7%; for location 69.5% and for organization 58%. The results using CRF are much better compared to those using ME because CRF discovers relevant features and also adds new features in the process. Li and McCallum[71] described the application of CRF with feature induction to the Hindi NER task. They considered three types of entities: person, location, and organization. They used the BIO format to demarcate entity boundaries,. The training corpus consisted of 340K words from the BBC (British Broadcasting Corporation) and EMI (External Machine Interface). There were 4,540 person, 7,342 location, and 3,181 organization NEs in the training data. A combination of Gaussian prior and early-stopping based on the results of tenfold cross-validation was used to reduce overfitting. Finally, the experimental results for validation and test sets were 82.55% and 71.50% respectively. The high accuracy was due to the use of Gaussian prior and early stopping. It is one of the simplest and most widely used means of avoiding overfitting and dividing the data into two sets: a training set and a validation set to reduce overfitting.

*HMM Approach*: Chopra et al.[23] discuss NER in the Hindi language by using

the rule-based approach and HMM approach. They showed that only a rule-based approach gave an F-measure of 47.5% whereas HMM gives 89.78% and a combination of both performs well in the NER based system.

### 2.2.3 NER in Telugu

*ME Approach*: Raju et al.[97] developed a Telugu NER system by using the ME approach. The corpus was collected from the *Eenaadu* and *Vaarta* newspapers and the Telugu Wikipedia. The system used contextual information for words such as suffixes and prefixes, frequent word lists, and bigrams. A gazetteer list was also prepared manually or semi-automatically from the corpus. They obtained F-measures of 72.07% for person, 60.76% for organization, 68.40% for location, and 45.28% for others, respectively. The reason for the low accuracy compared to the other languages was that Telugu, which is a Dravidian language, has a very complex morphology. Also, there was lack of a proper amount of annotated data, name dictionaries, and a good morphological analyzer.

*CRF Approach*: Srikanth and Murthy[125] used part of the Language Engineering Research Centre Telugu corpus consisting of a variety of books and articles at the University of Hyderabad (LERC-UoH). The authors also used two popular newspaper corpora, the *Andhraprabha* corpus consisting of 1.3 million words and the *Eenaadu* corpus consisting of 26 million words in which there were approximately 200K unique wordforms. The features used include morphological features, lengths of words, stop words, affixes, POS tags, orthographic information, and suffixes. A CRF-based noun tagger was built for noun identification. It was trained on a manually-tagged dataset of 13,425 words and tested on a test data set of 6,223 words. The authors obtained an F-measure of 91.95%. Then they developed a rule-based NER system using a corpus of 72,152 words including 6,268 NEs. The main

goal was to identify person, place, and organization names by using a bootstrapping approach. Finally, they also developed a CRF-based NER system as well. They achieved overall F-measures between 80% and 97% in various experiments. In the final experiment, the authors developed a CRF-based noun tagger whose output was one of the features for the CRF-based NER system. The results obtained were much better than the other approaches, as they developed training data using a heuristic-based system through bootstrapping.

Shishtla et al.[119] conducted an experiment on the development data released as a part of NER for the South and South-east Asian Languages (NERSSEAL 2008) Competition using CRF. The corpus consisted of 64,026 tokens of which 10,894 were NEs. The corpus was divided into training and testing sets where the former consisted of 46,068 tokens out of which 8,485 were NEs, and the latter consisted of 17,951 tokens of which 2,407 were NEs. Tagging of the corpus was done using the BIE format. The authors used language-independent features such as previous and next word information, prefixes and suffixes; and language-dependent features such POS tags, and chunks of words to help identify the boundary of NEs. The best-performing model gave an F-measure of 89.8%, which was comparable to the result obtained by Srikanth and Murthy[125].

### 2.2.4 NER in Tamil

*CRF Approach*: Vijaykrishna and Sobha[96] developed a domain-specific Tamil NER for tourism using CRF. It handled morphological inflection and nested tagging of NEs with a hierarchical tagset consisting of 106 tags. CRF++, an open-source toolkit for linear-chain CRF, was used to build a CRF model for NER. The attributes used to infer whether a phrase was an NE or not were roots of words, POS tags, patterns, and bigrams of NE labels. A corpus size of 94K was tagged manually for

POS, NP chunking, and NE annotations. The CRF was trained and CRF models for each of the levels in the hierarchy were obtained. The system obtained a Precision of 88.52%, Recall of 73.71% and F-measure of 80.44%. Precision was high as tagging was done only when the root word was taken from the training corpus, or the context of the current word was similar to the context of the NEs already in the training corpus. But when the NEs and their context were new to the CRF, they were most likely to not be tagged, resulting in low Recall. Malarkodi et al.[104] describe various challenges while developing Tamil NER using the CRF approach. A tagset of 106 was used for the approach. Various features were used and they discussed the ways to overcome diffeent challenges using morphological features, the smoothing technique, etc.

*Hybrid Approach*: Pandian et al.[92] presented a hybrid three-stage approach for Tamil NER. The first phase included the classification of the NEs by shallow parsing using a dictionary of word clues and case markers. In the second phase, shallow semantic parsing and syntactic and semantic information were used to identify the NE type. The final phase included statistical information from a training corpus. The Viterbi algorithm was used to identify the best sequence for the first two phases and then modified to resolve the problem of free-word order. Both NER tags and POS tags were used as the hidden variables in the algorithm. The system was able to obtain an F-measure of 72.72% for various entity types The accuracy of the NER was greatly dependent on the training corpus and the best results were obtained when both the test and the training corpora were similar.

### 2.2.5 NER In Oriya

Biswas et al.[17] presented a hybrid system for Oriya that applied ME, HMM and handcrafted rules to recognize NEs. First, the ME model was used to identify NEs

from the corpus. Then this tagged corpus was regarded as training data for the HMM which was used for the final tagging. The features used included suffix and prefix, root words, gazetteer features, POS, and morphological features. Linguistic rules were also used to identify NEs. The annotated data used in the system were in BIO[10]format. The system obtained an F-measure between 75% and 90%. Finally, the authors concluded that the performance of NER could be improved by using a hybrid approach compared to using only a single statistical model and that the developed system was adaptable to different domains regardless of the size of the training corpus.

### 2.2.6 NER in Urdu

*Rule-based Approach*: Riaz[100] discussed different approaches NER and challenges of NER, particularly for Urdu. The author identified the complex relationship between Hindi and Urdu and found that NER computation models for Hindi could not be used for Urdu NER. He also described a rule-based NER system which outperformed the models that used statistical learning. The experiment was carried out on the Becker-Riaz corpus consisting of 2,262 documents and the result showed an F-measure of 91.11%. He concluded that the rule-based approach was superior to the CRF approach used in the International Joint Conference on Natural Language Processing (IJCNLP) 2008 NER workshop. Jahangir et al.[57] discussed NER using the unigram and bigram models. The gazetteer list was also prepared for the recognition of NEs. The unigram NER tagger using the gazetteer list achieves Precision of 85.21%, Recall 88.63%, and F-measure 75.14%; the bigram gave Precision of 66.20%, Recall 88.18%, and F-measure 75.83%.

*HMM Approach*: Jahan and Siddiqui[56] discussed NER using the HMM model

[10]Beginning, Internal, and Outside words of an NE

and their system performs with 100% accuracy. Whereas Naz et al.[89] presented a brief overview of the previous work done on NER systems for South and South-east Asian Languages (SSEAL), the existing work in Urdu NER is a scarcely resourced and morphologically rich language.

*CRF Approach*: Mukund and Srihari[85] proposed a model that involved four levels of text processing for Urdu. Since they had limited data for training both the POS tagger as well as the NE models, a technique was proposed to help maximize the learning for efficient tagging. Two types of models were used. One a two-stage model that used POS information to perform NE tagging, and the other a four-stage model for NE tagging that used POS information with bootstrapping. CRF++[11], an open-source tool that used the CRF learning algorithm was used. The two-stage model achieved an F-measure of 55.3%; F-measure for the four-stage model was 68.9%. The better performance of the four-stage model was due to the use of a hybrid approach, i.e, the CRF approach and a rule-based approach, whereas the two-stage approach used only POS information to perform NE tagging.

### 2.2.7 NER in Punjabi

*CRF Approach*: Kaur et al.[61] developed a stand-alone Punjabi NER system using the CRF approach. Features such as context word features, prefixes and suffixes, POS, gazetteer lists, and NE information were used. Experimental results showed an F-measure of 80.92%. They found that the F-score varied with different NE tags, and concluded that the NER system could be changed according to the type of NE requirements. Gupta and Lehal[50] discuss NER in their paper for text summarization for Punjabi language using condition-based approaches. Various rules were developed such as prefix rule, suffix rule, proper-name rule, middle-name

[11]http://crfpp.sourceforge.net/

rule, and last-name rule, and a gazetteer was developed for prefix names, suffix names, proper names, middle names, and last names. The Precision, Recall, and F-Score for the condition-based NER approach were 89.32%, 83.4%, and 86.25% respectively. Singh[122] developed NER using the CRF approach. Various features like suffix, prefix, context words, POS, gazetteer along with unigram and bigram are used. Finally, the system gave an F-measure of 85.78%.

### 2.2.8 NER in Kannada

*Rule-based Approach*: Melinamath[80] discussed a rule-based approach for NER for Kannada by preparing a dictionary containing proper nouns, suffix list, prefix list, gazetteer, and a set of handcrafted rules and came with an F-measure of 85%-90%. On the other hand, Amarappa and Sathyanarayana[5] developed a supervised and rule-based NER system. The HMM model along with a set of rules are used to extract the root word of the NE, coming with an F-measure of 94.85%.
Besides these languages, work on NER in Odiya, Malayalam, Nepali and Manipuri can also be found. Balabantaray et al.[11] present an Odiya NER system using the CRF approach. Their approach handles nested tagging of NEs with a hierarchial tagset of 117. They perform tenfold cross-validation and their system gave an Fmeasure of 79%. Balabantaray[12] discuss an NER system based on CRF by using various types of features. Using the CRF++ tool the system gave an accuracy of 71%, but the performance of the system decreases when combined with the POS tag and a gazetteer. Jayan[58] presented a Malayalam NER system using two supervised taggers, TnT and SVM. They showed that for known words SVM performs better but for unknown words TnT performs well. However, for embedded tags a combination of rules along with TnT shows better results, giving an accuracy of 73.42%. Bindu and Idicula [103] presented a NE Recognizer for Malayalam

using the CRF method based on linguistic grammar principles and their system gave Precision of 85.52%, Recall 86.32%, and F-measure 85.61%. Singh et al.[124] discussed the development of the NER system in Manipuri. SVM along with the baseline system is used for the development of the system. Experimental results gave an F-measure of 94.59%. Bam and Shahi[13] discussed the NER in Nepali text using the SVM model. The system learns well from the small set of training data and increases the rate of learning when the training size is increased. Dey et al.[30] discussed the development of the NER detection tool using the HMM and rule-based approaches for Nepali language. Along with the NER tool they also developed the stemming tool, POS tool, etc.

### 2.2.9 International Work done on NER

Besides the work in Indian languages NER can also be found in other languages. Some of the work found in English, Arabic, Japanese, Chinese etc are briefly described below:

Benajiba et al.[15] discusses the problem of NER in Arabic using three machine learning approaches namely CRF,SVM and ME. Different types of features such as contextual, lexical and morphological and shallow syntactic features are considered. Two sets of data are used one of which is the standard ACE 2003 broadcast news data and the other is manually created data set. The ACE 2003 data defines four different classes:Person, Geographical and Political Entities, Organization and Facility. The system finally yields an highest F-measure of 83.34% using CRF on ACE data. Tran et al.[94] discusses NER in Vietnamese language using SVM which is then compared to CRF and found that SVM model outperforms CRF by optimizing its feature window size, thus obtaining an overall F-measure of 87.75%

whereas using CRF the system achieves an F-measure of 86.48%. Hwang et al.[134] discussesd NER for Korean language using HMM based NER using compound word construction principle. The compound word consists of proper noun, common noun or bound noun etc. These nouns are classifiied into 4 classes: independent entity, constituent entity, adjacent entity and not an entity. Experimental results show that this approach is better than rule-based in the Korean NER. Liao and Veeramachaneni[72] presented a semi-supervised algorithm for NER using CRF. The main aim of this approach is to select unlabelled data that has been classified with low confidence by the classifier trained on the original training data, but whose labels are known with high precision from independent evidence. The algorithm achieves an average improvement of 12 in recall and 4 in precision compared to supervised algorithm and it also showed that the algorithm achieves high accuracy when both the training and test sets are from different domains. Tkachenko and Simanovsky[129] presents a comprehensive set of features used in supervised NER. Different types of datasets are used such as CoNll 2003 which is an English language dataset for NER comprising Reuters newswire articles annotated with four entity types: Person, Location, Organization and Miscellaneous. OntoNotes version 4 is also an English dataset comprising Wall street Journal, CNN news and web blogs and lastly, NLPBA2004 is a bio-medical dataset. CRF approach is used which achieves an F-measure of 91.02% on CoNLL 2003 dataset and 81.4% on Onto Notes version 4 CNN dataset and NLPBA2004 dataset. Sasano and Kurohashi[111] use four types of structural information for Japanese NER using SVM. The four types are cache feature, coreference relations, syntactic feature and caseframe features. The approach is evaluated on CRL NE data and gives a higher F-measure than an existing approach that does not use structural information. They also conduct experiment on IREX NE data and an NE annotated web corpus and prove that structural information improves the performance of NER. Seok et al.[114] apply

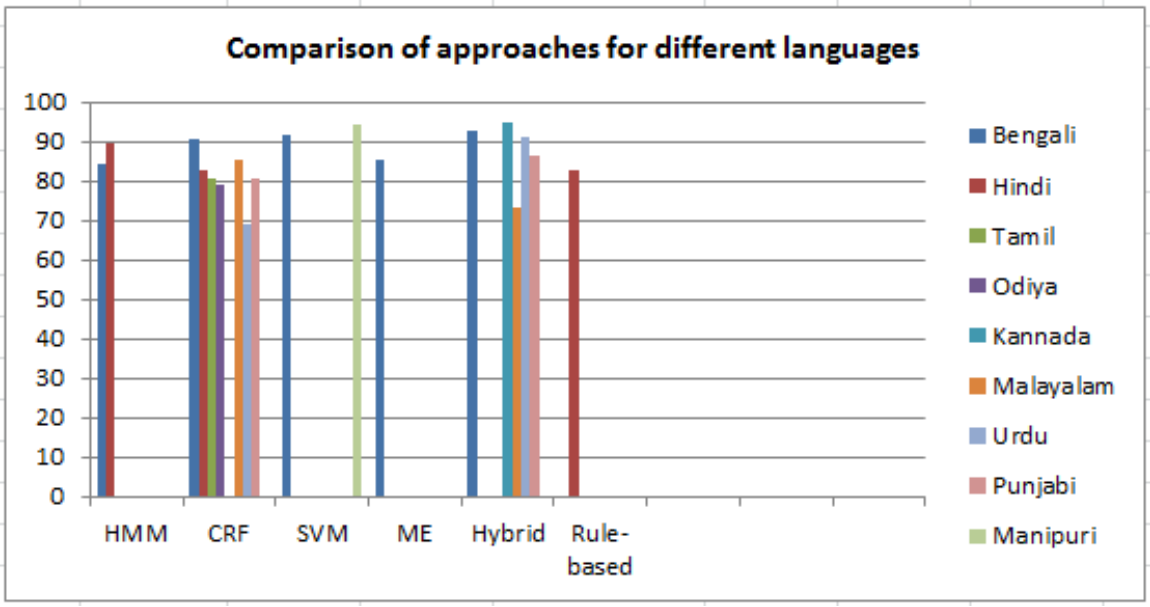
word embedding to feature for NER using CRF. They use GloVE, Word2 Vec, CCA as embedding method. The Reuters Corpus Volume 1 is used to create word embedding and 2003 shared taskcorpus of CoNLL is used for training and testing. Their experiment shows that CCA(85.96) in Test A and Word2 Vec(80.72) in Test B give the best performance but when word embedding is used as a feature for NER, it does not show better results than baseline that does not use word embedding. Florian et al.[44] present a framework for NER by combining four classifiers viz., robust linear classifier, maximum entropy, transformation based learning and Hidden Markov model. They use an English dataset which achieves an F-measure of 91.6% by combining all the four classifiers. Isozaki and Kazawa[54] in their paper first propose SVM based NE recognizer which shows that SVM classifiers are too inefficient. Thus they apply chunking and part-of-speech tagging which makes the system substantially faster.

### 2.2.10 Work done on NER using Gazetteer

Apart from the work done in NER using Machine learning approaches, work on NER can also be find using Gazetteer lists. Gazetteers are very useful resources for NER. The following are some of the work on NER using gazetteer list. Kozareva[65] presents the development of a spanish NER based on Machine-learning approach. The author explores the automatic generation of gazetteer lists from unlabeled data and building of NER with labelled and unlabelled data. CoNll 2002 data and the Spanish corpus are considered and several experiments are conducted to obtain a better idea for the performance of the classification method. Jahangir et al.[57] discuss Urdu NER using two basic n-gram namely unigram and bigram and also by using gazetteer list. The unigram NER tagger using gazetteer list achieves 65.25% Precision, 88.63% Recall and 75.14% F-measue and bigram NER

tagger using gazetteer list and backoff smoothing achieves Precison of 66.20%, Recall 88.18% and 75.83% F-measure. Saha et al.[109] decribe the approach for the preparation of gazetteer for NER in Indian languages. Two methodologies are used for the preparation of the gazetteer- one is the transliteration system which uses an intermediate alphabet and the second approach is the context pattern induction based domain specfic gazetteer preparation. The transliteration based approaches are useful when there is availability of English name list. The second approach uses bootstrapping to prepare the gazetteer from a large corpus starting from a few seed entities. Jahan and Chopra[55] discuss a hybrid approach for NER which is a combination of HMM and gazetteer to increase the accuracy of the NER system. Combining both the appraches they achieved an F-measure of 97.95% for Person and 98.80% for Location which is much better than the individual approach. By using HMM they achieve 85.70% for Person and 95.50% for location whereas 57% for Person and 37% for location using gazetteer. Dey and Prukayastha[31] present the gazetteer method and HMM using n-gram technique. They also describe the different approaches of NER and also the problem faced in handling Nepali grammar. Finally their experiment achieves an accuracy of 76.92% for Person, 88.14% for Organization and 77.42% for Location from 1000 sentences using n-gram and gazetteer method. Nadeau et al.[88] propose an NER system that combines named entity extraction with a simple form of named entity disambiguation. The performance of the unsupervised system is compared with that of base supervised system using MUC7 NER corpus. The unsupervised system is composed of two modules. The first one is used to create large gazetteer of entities such as list of cities, etc, and the second module uses simple heuristic to identify and classify entities in the context of a given document. Kazama and Torisawa[64] propose constructing a gazetteer for NER by using large scale clustering of dependency relations between verbs and multiword nouns(MN). They parallelize a clustering algorithm based

Figure 2.1: Comparison of approaches for different languages




on Expectation Maximization(EM) and thus enabling the construction of large scale MN clusters. IREX dataset is used to demonstrate the usefulness of cluster gazetteer . They also compare the cluster gazetteer with the wikipedia gazetteer by following the method of Kazama and Torisawa[63]. Further they demonstrate that the combination of the cluster gazetteer and a gazetteer extracted from wikipedia can further improve the accuracy in several cases.

The overall performance of the different approaches in different languages is shown in Fig 2.1. In the fig, we have considered only the best results for the different approaches and dataset more than 50K.

## 2.3 Tagsets

A standard tagset has been used by several authors for their work on NER in different Indian languages, namely, the IJCNLP-08 shared task tagset consisting of 12 tags [36]. There is a second tagset named Named Entity tagset defined by Ekbal and Bandyopadhyay consisting of 17 tags [32, 34, 35]. These two tagsets are described below.

### 2.3.1 The IJCNLP-08 Shared Task Tagset

This tagset includes the tags: NEP-Person name, NEL-Location name, NEO-Organization name, NED-Designation name, NEA-Abbreviation name, NEB-Brand, NETP-Title-person, NETO-Title-object, NEN-Number, NEM-Measure, NETE-Terms and NETI-Time.

### 2.3.2 The Named Entity Tagset by Ekbal and Bandopadhyay

This tagset contains the tags: PER: Single-word person name; LOC: Single-word location name; ORG: Single-word organization name; MISC: Single-word miscellaneous name; BIE-PER: Beginning, Internal or the end of a multiword person name; BIE-LOC: Beginning, Internal or the end of a multiword location name; BIE-ORGL: Beginning, Internal or the end of a multiword organization name; BIE-MISC: Beginning, Internal or the end of a multiword miscellaneous name; and NNE-words that are not NEs. In addition to these tagsets, several authors use the four most commonly used tags: PER: Person; LOC: Location;

ORG: Organization and MISC: Miscellaneous. [32] also used the 26 different POS tags given below. NN: Noun; NNP: Proper Noun; PRP: Pronoun; VFM: Verb Finite Main; VAUX: Verb Auxillary; VJJ: Verb Non-Finite Adjectival; VRB: Verb Non-Finite Adverbial; VNN: Verb Non-Finite Nominal; JJ: Adjective; RB: Adverb; NLOC: Noun location; PREP: Postposition; RP: Particle; CC: Conjunct; QW: Question Words; QF: Quantifier; QFNUM: Number Quantifiers; INTF: Intensifier; NEG: Negative Compound Nouns; NNC: Compound Common Nouns; NNPC: Compound Proper Nouns; NVB: Noun; JVB: Adj; RBVB: Adv; UH: Interjection words (HAM and interjections); and SYM: Special: Not classified in any of the above. NNP Proper Noun, PRP Pronoun, Besides this tagset, a tagset containing 106 tags Vijaykrishna and Sobha[96] has been used in Tamil where Level 1 contains 25 tags, Level 2 contains 50 tags and Level 3 has 31 tags.

### 2.3.3 CoNLL 2003 Shared Tagset

The tagset used by CoNLL 2003 consists of four tags, namely PER: Person; LOC: Location; ORG: Organization; and MISC: Miscellaneous. This tagging scheme is the IOB scheme originally put forward by Ramshaw and Marcus[99].

## 2.4 Evaluation Metrics used in NER

In information retrieval, *Precision* (also called positive predictive value) is the fraction of retrieved instances that are relevant, while *Recall* is the fraction of relevant instances that are retrieved. Both Precision and Recall are, therefore, based on an understanding measure of relevance. In a classification task, the Precision for a class is the number of true positives (i.e., the number of items correctly labeled

as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e., the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class). Recall is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e., the sum of true positives and false negatives, which are items which were not labeled as belonging to the positive class but should have been). In information retrieval contexts, Precision and Recall are defined in terms of a set of retrieved documents and a set of relevant documents.

Precision is the fraction of retrieved documents that are relevant to the find:

$$P = \frac{S_t \cap P_t}{P_t} \tag{2.1}$$

where $S_t$ = Relevant documents, and $P_t$ = Retrieved documents.

Recall in information retrieval is the fraction of the documents that are relevant to the query that are successfully retrieved.

$$R = \frac{S_t \cap P_t}{S_t} \tag{2.2}$$

where $S_t$ = Relevant documents, and $P_t$ = Retrieved documents. Individually Precision and Recall do not give a complete idea of effectiveness of a method. The standard approach is to use the F-measure, which is the harmonic mean of precision and recall, calculated as below:

$$F = 2\frac{P \times R}{P + R} \tag{2.3}$$

## 2.5 Cross-validation

Cross-validation is a method used for the evaluation of a supervised learning method. The model is given a set of known data (training data) and a set of unknown data (test data). In the cross-validation technique the data are divided into k subsets of which k-1 subsets are used for training data and then the learned program is tested on the one remaining subset. The process is repeated k times and is known as k-fold cross validation. In NER it is the most widely used technique for evaluation.
Cross-validation can be classified into two types: exhaustive, and non-exhaustive cross-validation.

1. *Exhaustive cross-validation*:- In exhaustive cross-validation, the learning program learns and is tested on all possible ways to divide the original sample into a training and a validation set. There are two cases.

    - Leave-p-out cross-validation: In Leave-p-out cross-validation (LpO CV) the method uses p observations as the validation set and the remaining observations as the training set.
    - Leave-one-out cross-validation:

      Leave-one-out cross-validation (LOOCV) is a particular case of leave-p-out cross-validation with p = 1.

2. *Non-exhaustive cross-validation*:- Non-exhaustive cross-validation methods do not compute all ways of splitting the original sample. It is an approximations of leave-p-out cross-validation.

## 2.6 Conclusion

This chapter discusses the work on NER in different Indian languages using different approaches. The performance of the individual work along with its accuracy is also discussed. Different tagsets are used in the work. Moreover, there are several reasons that lead to the poor performance of NER systems. Lastly we summarize all the work done on different languages using different approaches along with the accuracies obtained in Table 2.2 and Table 2.1.

We have seen that a considerable amount of work has been done in NER in different languages using different approaches. From the survey done on different work by different researchers, it has been found that there are several reasons that lead to the poor performance of a particular system such as the smaller quantum of training data, inability to handle unknown words, and lack of a dictionary or a lexicon in a particular language. Moreover, differences in the test set and the development set also decreases the accuracy of a system. Different language independent and dependent features also help in identifying the named entities. Lack of proper resources, lack of a good morphological analyzer, and lack of a proper dictionary also degrade the system performance. Rule-based methods generally fail in cases of exceptions, but ML approaches fail less drastically in exceptions. Rule-sets are generally not exhaustive, and do not cover ambiguous words. For example, the word *Shree* (শ্রী) acts both as a prenominal word and a person name. Whenever there exists a prenominal word the next word is a person name. But when we come across a sentence like (শ্রী ঘৰলৈ গল)— (*Shree gharaloi gol*, meaning Shree went home). In this case the word after (শ্রী) is not a person named entity, which violates the rule.

Table 2.1: NER approaches in Indian languages

| Language | Reference | Approach | Data Size(words) | F-measure(%) |
|---|---|---|---|---|
| Bengali | Banerjee et al.[14] | Margin Infused Algo | Train-112845 | 89.69 |
| | | | Test-38708 | |
| | Ekbal and Bandyopadhyay[39] | Classifier | Train-150K | 92.98 |
| | | | Test-30K | |
| | Ekbal and Bandyopadhyay[37] | Classifier | Train-220K | 89.03 |
| | | | Test-30K | |
| | Hasan et al.[51] | NER and POS tagger | Train-220K | 70.87 |
| | | | Test-30K | |
| | Ekbal and Bandyopadhyay[35] | SVM | Train-130K | 91.8 |
| | | | Test-20K | |
| | Ekbal and Bandyopadhyay[40] | SVM | Train-150K | 84.15 |
| | | | Test-50K | |
| | Hasanuzzaman et al.[52] | ME | Train-102,467 | 85.22 |
| | | | Test-20K | |
| | Ekbal and Bandyopadhyay[32] | HMM | Train-130K | 84.5 |
| | | | Test-20K | |
| | Chaudhuri and Bhattacharya[21] | n-gram | Train-20 million | 89.51 |
| | | | Test-100,000 | |
| | Ekbal et al.[41] | CRF | Train-130K | 90.7 |
| | | | Test-20K | |
| | Ekbal and Bandyopadhyay[38] | Classifiers | 35K | 85.32 |
| | Ekbal and Bandyopadhyay[34] | Shallow-Parser | Train-541,171 | P-75.40 |
| | | | Test-5000 sentences | L-72.30 |
| | | | | O-71.37 |
| | | | | M-70.13 |

Table 2.2: NER approaches in Indian languages

| Language | Reference | Approach | Data Size(words) | F-measure(%) |
|---|---|---|---|---|
| Telugu | Srikanth and Murthy[125] | CRF | Train-13,425 | 80-97 |
| | | | Test-6223 | |
| | Shishtla et al.[119] | CRF | Train-46068 | 44.91 |
| | | | Test-1795 | |
| | Raju et al.[97] | ME | eenadu, vaartha newspaper | P-72.07<br>O-60.76<br>L-68.40<br>Others-45.28 |
| Tamil | Vijaykrishna and Sobha[96] | CRF | 94K | 80.44 |
| | Pandian et al.[92] | HMM | 5500 words | 72.72 |
| Hindi | Saha et al.[106] | ME | Train-243K | 75.89 |
| | | | Test-25K | |
| | Saha et al.[108] | ME | Train-243K | 81.52 |
| | | | Test- 25K | |
| | Hasanuzzaman et al.[52] | ME | Train-452974 | 82.66 |
| | | | Test-50K | |
| | Goyal[47] | CRF | Train-19825 sentences | 58.85 |
| | | | Test-4812 sentences | |
| | Ekbal and Bandyopadhyay[32] | HMM | Train-150K | 84.5 |
| | Gupta and Arora[49] | CRF | Test-150 sentences | P-66.7<br>L-69.5<br>O-68 |
| | Li and Callum[71] | CRF | 340 K | 82.55 |
| | Chopra et al.[23] | Rule-based | | 47.5 |
| | | HMM | | 89.78 |
| Odiya | Biswas et al.[17] | ME and HMM | 40210 words | 75-90 |
| | Balabantaray et al.[11] | CRF | 277KB | 79 |
| Urdu | Riaz[100] | Rule-based | 2262 documents | 91.11 |
| | Jahangir et al.[57] | Unigram | Train-179,896 | 71.14 |
| | | Bigram | Test-4917 | 75.83 |
| | Jahan and Siddiqui[56] | HMM | 57 words | 100 |
| | Mukund and Srihari[85] | CRF | Train-146,466 | 68.9 |
| | | | Test-8000 | |
| Punjabi | Kaur et al.[61] | CRF | Train-45,975 | 80.92 |
| | | | Test-15000 | |
| | Gupta and Lehal[50] | Condition-based | 200003 | 86.25 |
| | Singh[122] | CRF | 17 K | 85.78 |
| Kannada | Melinamath[80] | Rule-based | 2423,4203,6537 | 85-90 |
| | Amarappa and Sathyanarayana[5] | HMM+Rule-based | Train- 10000 | 94.85 |
| | | | Test-130 | |
| Malayalam | Jayan et al.[58] | TnT and SVM | Train-10000(tokens) | 73.42 |
| | | | 150(tokens) | |
| | Bindu and Idicula[103] | CRF | Train-8000 sentences | 85.61 |
| | | | Test-2000 sentences | |
| Manipuri | Singh et al.[124] | SVM | Train-28,629 | 94.59 |
| | | | Test-4763 | |