Hybrid Approach for Marathi Named Entity Recognition

Nita Patil
SOCS
N. M. U. Jalgaon (MS)
India
nvpatil@nmu.ac.in

Ajay S. Patil
SOCS
N. M. U. Jalgaon (MS)
India
aspatil@nmu.ac.in

B. V. Pawar
SOCS
N. M. U. Jalgaon (MS)
India
bvpawar@nmu.ac.in

Abstract

This paper describes a named entity recognition system that combines hidden markov model, handcrafted rules, and gazetteers to recognize named entities in Marathi language. The objective of the system is to recognize twelve types of NEs from the Marathi text. Marathi is morphologically rich and inflectional language. The inflections in NEs are handled by using lemmatization. The difficulties of zero and poor probabilities caused due to the sparse data are handled using pseudo word replacement and smoothing techniques. Viterbi algorithm is used for decoding and word disambiguation. The performance of the system is improved using gazetteers and grammar rules.

Keywords: Named Entity Recognition, Marathi, HMM, Gazetteers, Rules

1 Introduction

Named Entity Recognition (NER) is information extraction task which can play significant role in many different natural language processing tasks such as information retrieval, machine translation, question answering systems etc. Predefined entities in text such as people, organizations, locations, events, expressions such as amount, percentage, numbers, date, time are named entities (NEs). Identification of NEs from unstructured text and their classification into suitable NE class is known as NER. This paper describes a hybrid model based on Hidden Markov Model (HMM), handcrafted rules and gazetteers to recognize named entities in Marathi. The difficulties of unseen probabilities are handled by pseudo word replacement and poor probabilities caused due to sparse data are handled using smoothing techniques. Viterbi alogirithm is used for decoding and word disambiguation. The performance of the system is improved using gazetteers. Linguistic rules are used to generate patterns that can recognize dates, time and numerical expressions. Following MUC specifications twelve types of NE are considered in recognition problem they are Person, Organization, Location, Miscellaneous, Amount, Number, Date, Time, Year, Month, Day and Measure. Patil (2017) reported the NER system based on trigram HMM model trained using pre-processed data for the Marathi language. The system uses Viterbi decoding to generate the optimal tag sequence for the test data. The system implemented using lemma model with trigram HMM has performed well in NE recognition, but it has further scope for improvement. Numerical NEs generally follow some fixed patterns, hence linguistic knowledge based recognition could be the better choice than probability based recognition. The study aims to improve NE recognition rate by combining effectiveness of statistical model with goodness of rule and gazetteer based technique for Marathi NER. The paper is organized in five main sections. Introduction and literature survey is discussed in first and second section. Supervised learning method for Marathi NER that uses HMM is described in third section. Fourth section briefs about rules and gazetteer based Marathi named entity recognition and the fifth section of the paper describes proposed hybrid model for development of Marathi NER system.

Proc. of ICON-2017, Kolkata, India. December 2017 ©*2017 NLPAI*, pages 103–111

2 Related Work

Research in named entity recognition for Indian languages is initiated by (Bandyopadhyay (2007), Varma (2008), Murthy (2008), Nusrat (2008), Bhattacharya (2009)). Many researchers have proposed rule based NER systems (Krupka (1998), William (1998), Awaghad (2009), Kashif (2010), Sasidhar (2011)) that give accurate results and achieve high performance. But the downside of this approach is lack of robustness and portability. Also, high maintenance is needed. Recently NER problems are solved by most of the researchers using statistical machine learning approach which uses mathematical and statistical models to train and test the data. Reasonable performance is reported by using this approach by the researchers (Daniel (1999), John (2001), GuoDong (2002), Asif (2008)). One more thought towards solving NER problem is combining the goodness of both approaches to achieve great performance and minimize the drawback is using Hybrid approach (Raymond (2006), Branimir (2008), Alireza (2008), Sitanath (2009), Xueqing (2009). Hybrid approach combines hand crafted rules with machine learning techniques. The time-consuming work like creation of resources can be done using machine learning and the other important task like preprocessing and post-processing can be done using hand crafted rules.

3 Machine Learning for NE Recognition

3.1 Using Hidden Markov Models

Hidden markov models relies on three parameters that are a matrix A of tag transition probabilities, a matrix B of emission or observation probabilities and a matrix π in which probability of the tag to occur in the initial state are recorded. Trigram HMM is defined as (K, V, λ), where $K = \{s_1, s_2, ..., s_n\}$ is a finite set of possible states, $V = \{x_1, x_2, ..., x_n\}$ is a finite set of possible observations and $\lambda = (\pi, A, B)$, where, $\pi = \{\pi_i\}$: Set of initial state probabilities and π_i : Initial probability that system starts at state i, $A = \{a_{ij}\}$: Set of state transition probabilities and a_{ij}: Probability of going to state j from state i, $B = \{b_i\{x_k\}\}$: Set of emission probabilities and $b_i\{x_k\}$: Probability of generating symbol x_k at state i. Maximum likelihood estimates are used to estimate parameters of λ model as, $a_{ijk} = \frac{C(i,j,k)}{C(i,j)}$ and $b_i\{x_k\} = \frac{C(i \rightsquigarrow x_k)}{C(i)}$. The start of the sentence is marked by $**$ and end of the sentence is marked by $STOP$ tag. The probability of state sequence $s_1, s_2.....s_{n+1}$ for given $x_1, x_2.....x_n$ observation sequence for NE tagging can be computed as,

$$P(x_1x_2.....x_n|s_1s_2.....s_{n+1}) \cong \prod_{i=1}^{n+1} q(s_i|s_{i-2}, s_{i-1}) \times \prod_{i=1}^{n} e(x_i|s_i)$$

Where q and e are parameters for maximum likelihood estimates. If we have $n = 6, x_1, x_2, ..., x_6$ equal to the sentence टप्प्यात १० हजार रुपये अनुदान., and $s_1, s_2, ..., s_7$ equal to the tag sequence O B-AMT I-AMT E-AMT O O STOP, then Bigram counts (Mat_{BC}) for probable tag sequence O B-AMT I-AMT E-AMT O O STOP for the sentence टप्प्यात १० हजार रुपये अनुदान. is,

$\text{Mat}_{\text{BC}} =$

	*	B-AMT	I-AMT	E-AMT	O	STOP
*	26462	44	0	0	19544	0
B-AMT	0	0	937	491	0	0
I-AMT	0	0	768	936	0	0
E-AMT	0	24	0	0	1335	1
O	0	1227	0	0	265391	26305
STOP	0	0	0	0	0	0

Unigram counts (Mat_{UC}) for probable tag sequence O B-AMT I-AMT E-AMT O O STOP is

$\text{Mat}_{\text{UC}} =$

*	B-AMT	I-AMT	E-AMT	O	STOP
0	1428	1705	1427	323621	0

Bigram probabilities (Mat_{BP}) for probable tag sequence O B-AMT I-AMT E-AMT O O STOP is,

$$\text{Mat}_{\text{BP}} = \begin{array}{c} \\ * \\ \text{B-AMT} \\ \text{I-AMT} \\ \text{E-AMT} \\ \text{O} \\ \text{STOP} \end{array} \begin{array}{c} \begin{array}{cccccc} * & \text{B-AMT} & \text{I-AMT} & \text{E-AMT} & \text{O} & \text{STOP} \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.656 & 0.344 & 0 & 0 \\ 0 & 0 & 0.450 & 0.549 & 0 & 0 \\ 0 & 0.017 & 0 & 0 & 0.936 & 0.001 \\ 0 & 0.004 & 0 & 0 & 0.820 & 0.081 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

The q and e parameter estimations for above sentence are

$$\begin{bmatrix} q(\text{O}|*) & = \frac{\text{C(*,O)}}{\text{C(*)}} & = 0.73857 \\ q(\text{B-AMT}|\text{O}) & = \frac{\text{C(O,B-AMT)}}{\text{C(O)}} & = 0.00379 \\ q(\text{I-AMT}|\text{B-AMT}) & = \frac{\text{C(B-AMT,I-AMT)}}{\text{C(B-AMT)}} & = 0.656162 \\ q(\text{E-AMT}|\text{I-AMT}) & = \frac{\text{C(I-AMT,E-AMT)}}{\text{C(I-AMT)}} & = 0.548974 \\ q(\text{O}|\text{E-AMT}) & = \frac{\text{C(E-AMT,O)}}{\text{C(E-AMT)}} & = 0.93553 \\ q(\text{O}|\text{O}) & = \frac{\text{C(O,O)}}{\text{C(O)}} & = 0.82007 \\ q(\text{O}|\text{STOP}) & = \frac{\text{C(O,STOP)}}{\text{C(O)}} & = 0.08128 \\ e(\text{टप्प्यात}|\text{O}) & = \frac{C(O \rightsquigarrow \text{टप्प्यात})}{\text{C(O)}} & = 0.000108 \\ e(\text{१०}|\text{B-AMT}) & = \frac{C(B-AMT \rightsquigarrow \text{१०})}{\text{C(B-AMT)}} & = 0.006303 \\ e(\text{हजार}|\text{I-AMT}) & = \frac{C(I-AMT \rightsquigarrow \text{हजार})}{\text{C(I-AMT)}} & = 0.226979 \\ e(\text{रुपये}|\text{E-AMT}) & = \frac{C(E-AMT \rightsquigarrow \text{रुपये})}{\text{C(E-AMT)}} & = 0.280308 \\ e(\text{अनुदान}|\text{O}) & = \frac{C(O \rightsquigarrow \text{अनुदान})}{\text{C(O)}} & = 0.000121 \\ e(.|\text{O}) & = \frac{C(O \rightsquigarrow .)}{\text{C(O)}} & = 0.079025 \end{bmatrix}$$

Bigram probability for an optimal tag sequence O B-AMT I-AMT E-AMT O O STOP for the sentence टप्प्यात १० हजार रुपये अनुदान., is,

$$\begin{aligned} P(x_1...x_6, s_1...s_7) = \\ & q(O|*) \\ & \times q(B-AMT|O) \\ & \times q(I-AMT|B-AMT) \\ & \times q(E-AMT|I-AMT) \\ & \times q(O|E-AMT) \\ & \times q(O|O) \\ & \times q(O|STOP) \\ & \times e(\text{टप्प्यात}|O) \\ & \times e(\text{१०}|B-AMT) \\ & \times e(\text{हजार}|I-AMT) \\ & \times e(\text{रुपये}|E-AMT) \\ & \times e(\text{अनुदान}|O) \\ & \times e(.|O) \\ & = 2.59683\times10^{-17} \end{aligned}$$

The probability of optimal tag sequence for a given word sequence is illustrated in above example. Similar probabilities are computed for all possible tag sequences for a given sentence using MLE estimation. Among all such possible tag sequences for a given sentence, the optimal path of tag sequence is to be selected. The tag sequence with highest probability is selected. This decoding is done by Viterbi algorithm(section 3.3). The trellis diagram for Viterbi decoding for a sample sentence टप्प्यात १० हजार रुपये अनुदान., is shown in figure 1.

3.1.1 Preprocessing Data

The lemmatization based technique (Patil (2017) is implemented in which inflected word forms are replaced by specialized tokens. Ontologies for number names in words, time, length, weight, electricity, temperature, area, volume and units of currency has been developed. The Marathi text is preprocessed using lemmatization based technique to deal with the inflections in named entities.

3.1.2 Minimizing Comparisons

Twelve different types of NEs using 40 tags need to be recognized by the NE recognizer. General trigram HMM assigns every tag to each word, computes bigram, trigram and unigram probabilities and assigns most probable

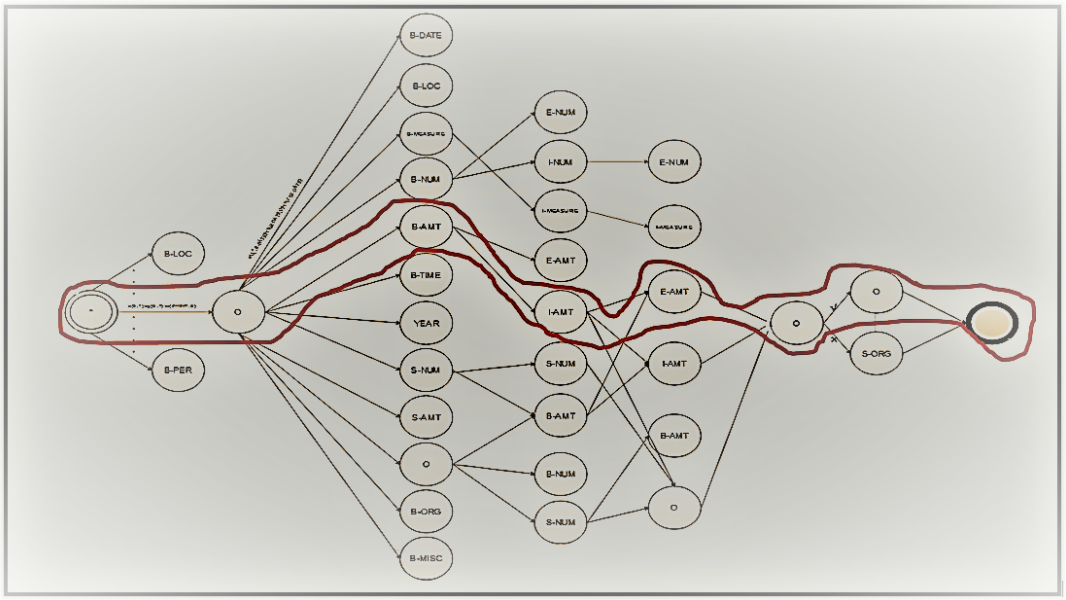

Figure 1: NE Tag Decoding

tag to the word based on maximum probability. We have taken two sentences to find number of trigram and emission probability computations. If 40 is the number of tags in tag set, then 40 tags are assigned to first word in sentence, 1600 bigram combinations are assigned to first and second word in sentence and 64,000 trigram combinations are assigned to first, second and third word. Thus, 64,000 trigram combinations are assigned to remaining words in sentence. Two * symbols are added to first word in sentence to make trigram. Trigram probability (T_P) is the ratio of trigram count (T_C) to bigram count (B_C) i.e. $T_P = T_C/B_C$. Combination B_C that is not seen in training becomes zero, zero value at denominator results in infinite trigram probability. The difficulty introduced because of unseen B_C is solved by using two solutions. First solution is return value 1 for unseen B_C which can temporary solve the problem. Second solution is find out all the bigram combinations which are never seen in training as well as not expected during testing. All such combinations are called invalid bigram combinations. There are approximately 1089 bigram tag combinations that are never seen in training and not expected in testing some of them are shown in table 1. For all combinations, which are invalid computation of T_P is skipped so that load of algorithm execution can be released to some extent as well wrong trigram state assignment to observations can be controlled. Comparison between first and second solution is shown in table 2.

CurTag	NextTag	CurTag	NextTag
B-LOC	B-TIME	B-LOC	I-DATE
B-LOC	B-AMT	B-LOC	I-MEAS
B-LOC	B-DATE	B-LOC	I-MISC
B-LOC	B-LOC	B-LOC	I-NUM
B-LOC	B-MEAS	B-LOC	I-ORG
B-LOC	B-MISC	B-LOC	I-PER
B-LOC	B-NUM	B-LOC	MONTH
B-LOC	B-ORG	B-LOC	O
B-LOC	B-PER	B-LOC	S-TIME
B-LOC	E-TIME	B-LOC	S-AMT
B-LOC	E-AMT	B-LOC	S-DATE
B-LOC	E-DATE	B-LOC	S-LOC
B-LOC	E-MEAS	B-LOC	S-MEAS
B-LOC	E-MISC	B-LOC	S-MISC
B-LOC	E-NUM	B-LOC	S-NUM
B-LOC	E-ORG	B-LOC	S-ORG
B-LOC	E-PER	B-LOC	S-PER
B-LOC	I-TIME	B-LOC	WEEKDAY
B-LOC	I-AMT	B-LOC	YEAR

Table 1: Part of Unseen Bigram Tag Combinations

T_P Computations	Solution 1	Solution 2
Trigram comparisons	142596	36581
Non zero T_Ps	3959	3959
Zero T_Ps	138637	32622

Table 2: Comparison between Tp Computations for Two Solutions

3.2 Viterbi Decoding

Viterbi algorithm is used to predict most likely tag sequence for an input sequence. The algorithm finds most probable state sequence $s_1, s_2, ..., s_n$ for a observation sentence $x_1, x_2, ..., x_n$. The problem of maximizing $P(s_1, s_2...s_n|x_1, x_2.....x_n)$ is addressed using $argmax_{s_1....s_n} P(s_1 s_2.....s_n|*, x_1 x_2.....x_n, STOP)$.

3.3 Handling Unseen words

Unseen words are absent in training, therefore their prediction probability becomes zero. If frequency of observation in test set is less than or equal to 5, then that observation is treated as rare word. Non frequent words in test set are replaced by $< RARE >$ token. Katz back-off smoothing is used to estimate the count of words that are never seen in training.

4 Linguistics for NE Recognition

Linguistic knowledge to recognize Marathi NEs is represented using indicator word lists, gazetteers, and grammar rules. This subsection provides brief information about the linguistic resources developed for detection of NEs from newspaper articles.

4.1 Indicator Word Lists

The indicators often surrounding the NEs can act as trigger words in identification of NEs in their context. Such words play significant role in designing heuristics to indicate NEs within the text. Certain words exist in text that are not indicators but are ambiguous NEs and must be treated separately. The word lists for indicators such as title person, awards, degree, person name suffixes, suffixes to person first name, suffixes to person last name, collision of proper and common noun, collision of proper, common noun and verbs, ambiguous last names, Marathi abbreviations, English in Devanagari abbreviations, location indicators, location suffixes etc. were developed to assists NE recognition by rule based NER algorithm.

4.2 Using Gazetteers

Gazetteer for first names, last names, organizations names, miscellaneous names, days of the week, month names (English and Marathi), single word location, organization and miscellaneous etc. were created. The word form(s) which is (are) untagged if found in some gazette(s), then the appropriate tag(s) is (are) assigned to the word form(s) based on the gazette(s) in which it found.

4.3 Using Grammatical Rules

The grammatical rules are a set of grammatical patterns designed to derive NEs based on lemmatization. Grammatical patterns were indicated using regular expressions. Several rules have been developed, which are used to extract person, location, amount, measure, date, time, and number entities.

5 Experimental Work

5.1 Dataset Preparation

FIRE-2010 corpus is used to develop NE annotated corpus by manually tagging 12 types of NEs. 27,177 sentences of Marathi text have been annotated using IOBES scheme. Training data developed for Marathi NER consists of 4,01,295 word forms that comprise of 12,303 person names, 7,440 organization, 10,015 location, 3,242 miscellaneous, 7,093 number, 1,500 amount, 2,967 measure, 1,549 date, 369 time, 197 month, 456 weekdays, and 395 year named entities. The rich morphology of the Marathi language allows adding suffixes and prefixes to a morpheme to add semantic to a word and to create meaningful context. It is observed during corpus annotation that almost all NE instances are present in inflected form. Although the dataset is large enough, frequency count of word is found to be lower since inflections result in same word appearing in different forms. This further results in poor probabilities and sparse data problem in MLE estimates. Lemmatization based preprocessing deals with such inflections and is used in the preprocessing of training and testing datasets.

5.1.1 Held Out Test Dataset Preparation

Two sets of training and testing datasets is created by dividing the NE annotated corpus preprocessed using lemmatization in 80:20 and 90:10 percent proportions. The actual number of sentences in the corpus are computed, 20% of the total sentences in the corpus were randomly selected and removed from the corpus. The set of randomly selected sentences

NE Class	NE Annotated Data	Training Dataset1	Held-out Dataset1	Training Dataset2	Held-out Dataset2
Person	12,303	11,998	0305	12,285	018
Organization	07,440	07,236	0204	07,421	019
Location	10,015	09,723	0292	09,983	032
Miscellaneous	03,242	03,170	0072	03,231	011
Number	07,093	06,893	0200	07,081	012
Amount	01,500	01,463	0037	01,494	006
Measure	02,967	02,887	0080	02,958	009
Date	01,549	01,515	0034	01,541	008
Time	00369	00360	0009	00363	006
Month	00197	00193	0004	00190	007
Weekday	00456	00441	0015	00455	001
Year	00395	00384	0011	00389	006
Total NEs	47,526	46,263	1,263	47,391	135
#Sentences	27,177	26,462	0715	27,127	050

Table 3: Held Out Training and Testing Dataset Details

is termed as Held-out dataset1. The remaining sentences (80%) in the corpus (training dataset1) were used to train the NER system. Similarly, 10% of the total sentences in the corpus were randomly selected, removed and stored in Held-out dataset2. The remaining sentences (90%) in the corpus (training dataset2) were used to train the NER system. The total number of NE instances found in the training dataset1, training dataset2, held-out dataset1 and held-out dataset2 are shown in table 3.

NE Class	Train1	Unseen1	Unseen2
Person	11,998	33	08
Organization	07,236	15	16
Location	09,723	17	22
Miscellaneous	03,170	16	02
Number	06,893	10	16
Amount	01,463	05	01
Measure	02,887	02	06
Date	01,515	03	03
Time	00,360	01	01
Month	00,193	02	01
Weekday	00,441	01	01
Year	00,384	04	04
Total NEs	46,263	109	81
Sentences	26,462	33	24

Table 4: Unseen Test Dataset Details

5.1.2 Unseen Test Dataset Preparation

Unseen dataset1 is a dataset composed of news items taken from online eSakal newspaper in October 2016. Unseen dataset2 is a dataset composed of news items taken from online eSakal newspaper in February 2017. Both the unseen datasets were tokenized and preprocessed using lemmatization. The total number of NE instances found in the unseen dataset1 and unseen dataset2 is shown in table 4. The NE annotated corpus pre-processed using lemmatization consisting of 27,177 sentences mentioned in the dataset preparation section is used to train the NER system.

5.2 NER System Architecture

The proposed NER system applies statistical algorithm i.e. trigram HMM using lemmatization algorithm to test data. This algorithm recognizes Marathi NEs satisfactorily. It also deals with unknown words and performs word disambiguation to some extent. There is possibility that some NEs might be untouched by the system. Therefore, rule and gazetteer based NER algorithm is cascaded to the NER system. The rule based algorithms do not modify the recognition carried by statistical algorithm, rather it tags only the untagged NEs in the test data. The NEs which are not contained in any gazetteer are termed as unseen NEs. The problem of unseen NEs is solved by statistical algorithm using pseudo word replacement. Therefore, continuous expansion of gazetteers is not required. Expected performance of the Marathi NE recognition is achieved using combining the statistical algorithm with the rule based algorithm. The architecture of NER system for the Marathi language that combines statistical named entity recognition, gazetteers and grammar rules is

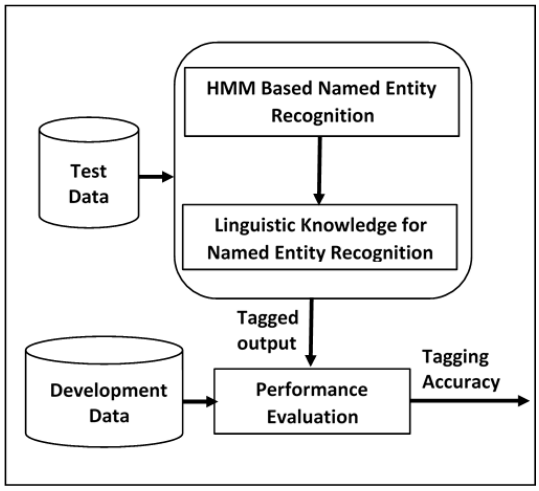

Figure 2: Marathi NER System

shown in figure 2.

5.3 Evaluation of Hybrid NER System

The performance of the Marathi NER based on hybrid approach is evaluated using four varying size datasets containing varying number of NEs. Out of them two datasets were held out and remaining two datasets were unknown datasets.
The system is trained on dataset(s) preprocessed using lemmatization. The performance of the system using held out datasets is shown in table 5 and 6. The overall NE identification accuracy reported by the system for held out dataset1 and 2 is 93.35% and 98.14% respectively. The average NE classification accuracy reported is 95.24% and 97.79% respectively.

The overall NE identification accuracy reported by the system for unseen dataset1 and 2 is 81.37% and 83.33% respectively which is relatively satisfactory. The average NE classification accuracy reported for unseen dataset1 and 2 is 83.09% and 84.23% respectively. The NE recognition accuracy for organization NE is relatively less result in unsatisfactory average NE classification accuracy for unseen dataset2. Numeric NEs in this dataset were accurately recognized than the enamex type of NEs by the system. The performance of the system using unseen datasets is shown in table 7 and 8 respectively. Overall NE identification accuracy and average NE classification accuracy is shown in graph 3 and 4 respectively.

NE Class	Precision	Recall	F1-Score
NEI	92.79	93.92	93.35
Person	84.05	86.35	85.18
Organization	95.02	98.96	96.95
Location	97.26	97.26	97.26
Miscellaneous	95.83	95.83	95.83
Number	96.43	90.43	93.33
Amount	80.00	100.0	88.89
Measure	100.0	100.0	100.0
Date	93.67	97.37	95.48
Time	81.82	100.0	90.00
Month	100.0	100.0	100.0
Weekday	100.0	100.0	100.0
Year	100.0	100.0	100.0
NEC	93.67	97.18	95.24

Table 5: NER System Performance on Held-out Dataset1

NE Class	Precision	Recall	F1-Score
NEI	98.51	97.78	98.14
Person	94.74	100.0	97.30
Organization	100.0	100.0	100.0
Location	100.0	96.88	98.41
Miscellaneous	100.0	100.0	100.0
Number	92.31	100.0	96.00
Amount	100.0	100.0	100.0
Measure	100.0	100.0	100.0
Date	100.0	100.0	100.0
Time	100.0	83.33	90.91
Month	100.0	100.0	100.0
Weekday	100.0	100.0	100.0
Year	100.0	83.33	90.91
NEC	98.92	96.96	97.79

Table 6: NER System Performance on Held-out Dataset2

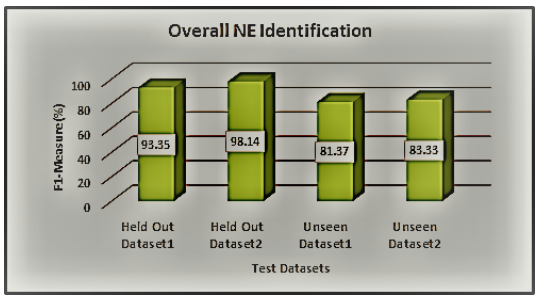

Figure 3: Overall NE Identification

NE Class	Precision	Recall	F1-Score
NEI	86.46	76.85	81.37
Person	78.57	66.67	72.13
Organization	90.91	66.67	76.93
Location	100.0	94.12	96.97
Miscellaneous	100.0	87.50	93.33
Number	69.23	90.00	78.26
Amount	66.67	80.00	72.73
Measure	100.0	50.00	66.67
Date	100.0	100.0	100.0
Time	100.0	100.0	100.0
Month	100.0	100.0	100.0
Weekday	100.0	100.0	100.0
Year	100.0	25.00	40.00
NEC	92.12	80.00	83.09

Table 7: NER System Performance on Unseen Dataset1

NE Class	Precision	Recall	F1-Score
NEI	92.31	75.95	83.33
Person	83.33	62.50	71.43
Organization	87.50	43.75	58.33
Location	85.00	77.27	80.95
Miscellaneous	100.0	0	0
Number	100.0	100.0	100.0
Amount	100.0	100.0	100.0
Measure	100.0	100.0	100.0
Date	100.0	100.0	100.0
Time	100.0	100.0	100.0
Month	100.0	100.0	100.0
Weekday	100.0	100.0	100.0
Year	100.0	100.0	100.0
NEC	96.32	81.96	84.23

Table 8: NER System Performance on Unseen Dataset2

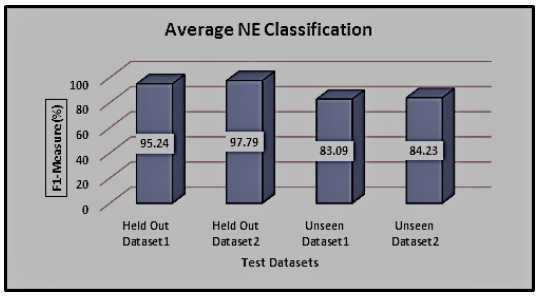

Figure 4: Overall NE Classification

The cumulative performance of Marathi NER system based on Hybrid approach for held out and unseen test datasets is shown in table 9. NE identification and classification reported by this system is 90% approximately, which is satisfactory for Marathi language.

Test Datasets	NEI	NEC
Held-Out Dataset1	93.35	95.24
Held-Out Dataset2	98.14	97.79
Unseen Dataset1	81.37	83.09
Unseen Dataset2	83.33	84.23
Average	89.05	90.09

Table 9: Average Performance of NER

6 Conclusion

A NER system for Marathi language is described that applies hidden markov model, language specific rules and gazetteers to the task of named entity recognition (NER) in Marathi language. Starting with named entity (NE) annotated corpora and lemmatization first a baseline NER system was implemented. Then some language specific rules are added to the system to recognize some specific NE classes. Also, some gazetteers and context patterns are added to the system to increase the performance. After preparing the one-level NER system, a set of rules are applied to identify the nested entities. The system can recognize 12 classes of NEs with 89.05% accuracy in average NE identification and 90.09% accuracy in average NE classification for held out and unseen test datasets in Marathi.

Acknowledgement

This research work is supported by grants under Rajiv Gandhi Science and Technology Commission, Govt. of Maharashtra, India.

References

Asif Ekbal and Sivaji Bandyopadhyay. 2007. *A Hidden Markov Model Based Named Entity Recognition System: Bengali and Hindi as Case Studies*. Springer International Conference on Pattern Recognition and Machine Intelligence (PReMI 2007) Heidelberg, LNCS, 4815:545–552.

Anup Patel, Ganesh Ramakrishnan and Pushpak Bhattacharya. 2009 *Incorporating Linguistic Expertise using ILP for Named Entity Recognition in Data Hungry Indian Languages.* In Proceedings of the 19th International Conference on Inductive Logic Programming (ILP'09), Leuven, Belgium,178–185.

Sudha Morwal, and Nusrat Jahan. 2013. *Named entity recognition using hidden markov model (hmm): An experimental result on Hindi, urdu and marathi languages.* International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE),3(4):671–675.

Praneeth Shishtla, Karthik Gali, Prasad Pingali, and Vasudeva Varma. 2008. *Experiments in Telugu NER: A Conditional Random Field Approach.* In Proceedings of the Workshop on NER for South and South East Asian languages (IJCNLP-08), Hyderabad, India,105–110.

P. Srikanth and Kavi Narayana Murthy. 2008. *Named Entity Recognition for Telugu.* In Proceedings of the Workshop on Named Entity Recognition for South and South East Asian Languages, Third International Joint Conference on Natural Langauge Processing (IJCNLP-08), Hyderabad, India, 41-50.

Krupka, G.R., and Hausman, K. 1998. *IsoQuest Inc: Description of the NetOwl Text Extraction System as used for MUC-7.* In Proceedings of Seventh Message Understanding Conference (MUC-7), Fairfax, Virgina.

William J Black, Fabio Rinaldi and David Mowatt. 1998. *Facile: Description Of The NE System Used For Muc-7.* In Proceedings of Seventh Message Understanding Conference (MUC-7), Fairfax, Virgina.

Awaghad Ashish Krishnarao, Himanshu Gahlot, Amit Srinet and D. S. Kushwaha. 2009. *A Comparative Study of Named Entity Recognition for Hindi using Sequential Learning Algorithms.* International Advance Computing Conference (IACC 2009), Patiala, India:1163-1168.

Kashif Riaz. 2010. *Rule-based Named Entity Recognition in Urdu.* In Proceedings of the 2010 Named Entities Workshop, ACL 2010, Uppsala, Sweden:126–135.

B. Sasidhar, P.M. Yohan, A. Vinaya Babu, A. Govardhan. 2011. *Named Entity Recognition in Telugu Language using Language Dependent Features and Rule based Approach.* International Journal of Computer Applications, 22(8):30-34.

Daniel M. Bikel, Richard L. Schwartz, and Ralph M. Weischedel. 1999. *An Algorithm that Learns What's in a Name.* Machine Learning, 34(1): 211-231.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data* In Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001):282-289.

GuoDong Zhou Jian Su. 2002 *Named Entity Recognition using an HMM-based Chunk Tagger.* In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, Pennsylvania:473-480.

Asif Ekbal and Sivaji Bandyopadhyay. 2008. *Bengali Named Entity Recognition Using Support Vector Machine.* In Proceedings of the Workshop on Named Entity Recognition for South and South East Asian Languages, Third International Joint Conference on Natural Langauge Processing (IJCNLP-08), Hyderabad, India: 51-58.

Raymond Chiong and Wang Wei. 2006. *Named Entity Recognition Using Hybrid Machine Learning Approach.* 5th IEEE International Conference Cognitive Informatics, (ICCI-2006), Volume 1:578-583.

Branimir T. Todorovic, Svetozar R. Rancic, Ivica M. Markovic, Edin H. Mulalic and Velimir M. Ilic. 2008. *Named Entity Recognition and Classification Using Context Hidden Markov Model.* 9th Symposium on Neural Network Applications in Electrical Engineering, NEUREL 2008, Belgrade, Serbia :43-46.

Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. 2008. *Name Entity Recognition Approach.* International Journal of Computer Science and Network Security, 8(2):320-325.

Sitanath Biswas, S. Mohanty, S.P. Mishra. 2009. *A Hybrid Oriya Named Entity Recognition System: Integrating HMM with MaxEnt.* In Proceedings of 2nd International Conference Emerging Trends in Engineering and Technology (ICETET 2009), Nagpur:639-643.

Xueqing Zhang, Zhen Liu, Huizhong Qiu, Yan Fu. 2009. *A Hybrid Approach for Chinese Named Entity Recognition in Music Domain.* In Proceedings of Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing , Chengdu, China:677-681.

Nita Patil, Ajay Patil and B. V. Pawar. 2017. *HMM based Named Entity Recognition for inflectional language.* IEEE International Conference on Computer, Communications, and Electronics (COMPTELIX 2017):565-572.