Generic Feature Selection Methodology to Named Entity Detection from Indian and European Languages

Chithamparanathan Saroja MALARKODI, Sobha Lalitha DEVI
AU-KBC Research Centre, MIT Campus of Anna University, Chromepet, Chennai, India
sobha@au-kbc.org

***Abstract*—This paper describes the development of language and domain independent Named Entity Recognition (NER) system which can identify named entities from any given dataset irrespective of the language and domain. The main novelty of the present work is the generic feature selection methodology which has been applied to 7 Indian languages and 5 European languages. The generic feature selection methodology was done in two ways; first using frequency based approach; secondly k-means++ clustering algorithm was used to validate the patterns obtained in the frequency based approach. The dataset used for the experiments belongs to different genre. To the best of our knowledge we are the first to work on the development of cross-lingual Named Entity (NE) system with 12 languages belongs to different language families. We have done the 10-fold cross validation and the system output has been analyzed for all the languages and causes of error cases was discussed in the error analysis section. The performance of our system is also compared with the existing systems.**

***Index Terms*—computational linguistics, natural language processing, machine learning, classification algorithms, supervised learning.**

I. INTRODUCTION

In order to make wise decisions, nowadays organizations, academic institutions and business people required the efficient text mining tools to extract the knowledgeable information from the wide pool of unstructured web data. The huge amount of unstructured data lies on the web, promotes the requirement of text mining and Natural Language Processing (NLP) tools. Text mining intends to discover the potential knowledge from huge datasets. Information Extraction (IE) is an important aspect of text mining, since it extracts the structured information from unstructured data by recognizing Named Entities (NEs) and their associated relations. IE consists of two important components, i) identification & classification of named entities and ii) relation extraction. NER is the task of recognizing and classifying the references of entities such as names of person, location, organization, product etc., in the given text. Extraction of named entities is an essential step to obtain intelligent information for NLP applications like Information Extraction, Machine Translation, Sentiment analysis, On-line Customer Support, Question & Answering (Q&A), Text Summarization, Search Algorithms etc...

Initially the term NER was defined in Message Understanding Conference (MUC) [1]. Nadeau et al. [2] has reported fifteen years of research carried out in the field of entity recognition. A maximum entropy based NER model has been developed for English & Spanish with good accuracy [3]. In 2002, CONLL shared task about NER was focused on Spanish and Dutch. The highest f-score obtained by participants for Spanish and Dutch are 81.39%, 77.05% respectively [4]. The CONLL 2003 offered dataset for English and German [5]. In that shared task, Florian et al. [6] achieved a highest accuracy for English 88.76% and German 72.41% using maximum entropy. Memory based learner has been introduced in order to find the names in German and English text [7]. Desmet and Hoste [8] created fine grained NE corpus for Dutch which consists of 6 NE types person, location, organization, product, events and miscellaneous. They have developed the Dutch NE system using classifier ensemble approach. The three classifiers used in the system development are Memory-based learning (MBL), Conditional Random Fields (CRF) and Support Vector Machines (SVM). These classifiers were trained on the features and the resulted outcome were ensemble using various voting mechanisms in genetic algorithm. Their classifier achieved the F-score of 84.91%. Varga et al. [9] developed a system to identify NEs using maximum entropy approach, named “hunner”. The Szeged NER corpus was used for training and evaluation purpose. It consists of 220k token subset of Szeged Corpus. Their system achieved 94.77% for the test set results. Szarvas et al. [10] developed a NE classifier for Hungarian & English using AdaBoostM1 and C4.5 decision tree algorithm. The outcome of the classifier was handled through voting mechanism. The CONLL 2003 dataset and Szeged corpus was used for the experiments. They achieved 91.41% for English and 94.77% using for Hungarian. Eleven teams have participated in the GermEVAL NER shared task. The NE tagged corpus given in the task has been collected from German Wikipedia and various online news articles. The GermEVAL NE dataset was annotated with 12 NE tags [12].

Many researchers have used several techniques in Indian languages. NERSSEAL shared task of IJCNLP has concentrated on Indian languages like Hindi, Bengali, Oriya, Telugu and Urdu. The dataset consists of 12 tags. The main techniques used by the various research groups are Conditional Random Fields (CRFs), Support Vector Machine (SVM), Maximum Entropy Markov Model (MEMM) and rule based approaches [13]. Saha et al. [14] worked on the feature reduction approaches for Hindi and Bengali. The various types of reduction techniques used are feature selection, feature extraction and feature clustering.

Digital Object Identifier 10.4316/AECE.2019.01011

The feature selection and evaluation metrics reported here are term frequency, TFiDF, information gain, pointwise mutual information, chi-square, statistic, NE class association metric (proposed by them). The various types of feature selection strategies are filter, wrapper and embedded. The empirical results show that the feature reduction methodology improves the overall performance of the system. Gupta et al. [15] presented the MEMM based approach which combines the Local and Global Information for NE Identification (CLGIN) to integrate the global characteristics of the corpus with the local context. The two main steps in CLGIN are as follows 1) Named Entity Identification (NGI) used to combines the global characteristics along with the language cues. The NE list generate from step 1 are given as a feature to next step. 2) the NE list from the previous steps are considered as a feature for MEMM. The dataset created from Gyaan Nidhi book, was tested with baseline MEMM, NGI and CLGIN. The results obtained showed that CLGIN approach performs better than other two approaches. Patil et al. [16] discussed the challenges of entity identification in Marathi. Kaur et al. [17] scored 84.19% F-M for Punjabi entity identification using context word feature in window of 3, 5, 7 and digit features using CRF. Ekbal et al. [18] contributed NER systems for Hindi & Bengali using CRF framework. The system was developed using 2 methods, one using language independent features and another using language specific features. The corpus used for their work was obtained from IJCNLP-08 NER shared task. The Bengali NE dataset consists of 1,22,467 tokens and Hindi dataset consists of 5,02,974 tokens. The tags which indicate person, location, organization, numerical, time & measurement expressions were used for the experiments. The empirical results shows that the system developed using language specific features yielded better results for both Bengali and Hindi. Kumar et al. [19] built NE systems for Hindi & Marathi using the Bisecting k-means clustering algorithms.

Bindu and Sumam Mary [20] used CRF based approach for Malayalam text. Raju et al. reported a NER system in Telugu language using maximum entropy [21]. The 3 phase hybrid NE recognizer was constructed for Tamil with six NE tags by Pandian et al. In statistical processing phase E-M algorithm is used with the initial probabilities obtained from the shallow & semantic parsing phase. They have claimed average f-score as 72.72% for various entity types [22]. Vijayakrishna and Sobha [23] focused on the Tamil NER for tourism domain which consists of nested tagging of named entities. They have used the hierarchical tag set of 106 tags. The various features used are root word, PoS, combination of word & PoS and NE gazetteers. Malarkodi and Sobha [24] built a NE system for Indian languages like Tamil, Telugu, Hindi, Marathi, Punjabi and Bengali. They reported the results obtained for language independent and dependent features. Malarkodi et al. [25] reported the challenges of named entity identification in Tamil and presented a system which overcame some of those issues.

Sobha et al. [26] has participated in the ICON NER Tool contest. They have used linguistic, word level, and orthographic features for CRF language model which is followed by post-processing rules. Gayen et al. [27] also submitted their test runs for ICON 2013 NER contest. The machine learning technique used in their work was Hidden Markov Model (HMM). In 2013 AU-KBC has organized NER shared task as part of Forum for Information Retrieval for Evaluation (FIRE), to create a benchmark data for Indian Languages. The dataset was released for 4 Indian Languages like Bengali, Hindi, Malayalam, and Tamil and also for English. Five teams have participated in the task. The various techniques used by the participants are CRF, rule based approach and list based search [28]. The 2nd edition of NER track for Indian Languages (IL) has organized as part of FIRE 2014 for English and 3 IL namely Hindi, Malayalam, and Tamil. The main focus of this track is nested entity identification. The participants have used CRF and SVM for system development [29]. Abinaya et al. [30] achieved the highest score in the FIRE 2014 NER shared task. The feature reduction approaches based on clustering and feature reduction techniques has been applied for bio-medical domain using maximum entropy classifier [31].

The feature selection methodology used in this work is validated using K-means clustering based approach. The K-means algorithm randomly divides the dataset into k groups and computes the distance between each data point and the cluster head. Then allot the data point to the nearest cluster head using the Euclidian distance measure. Zahra et al. [32] presented the comparative study of various centroid selection approaches and also experimented the performance and cost consumption of each approaches. They proposed a k-means clustering based recommendation algorithm to solve to scalability issues occurring in traditional recommender systems which lead to the inaccurate recommendations and increases the cost for cluster training. In order to resolve these issues encountered in traditional k-means systems, they introduced a centroid selection method which reduces the cost consumption and also improve the performance. The experimental results show that the presented solution gives a better quality cluster converges faster than the previous works. The traditional k-means clustering leads to the unbalanced boundaries to some clusters. In order to provide the proper lower and upper bounds to the clusters Zahng et al. [33] introduced an improved rough k-means algorithm based on the weighted distance measure using Gaussian function. In traditional k-means algorithm the same weight is assigned to the objects exists in the lower boundary region. But in the proposed algorithm the new weighted distance measure using Gaussian function is assigned to the data point in the lower boundary regions. Thus by the proposed algorithm provides the balanced boundaries to the given dataset. Borlea et al. [34] introduced a new centroid update approach to reduce the number of iterations in the classical k-means clustering. Based on the S-divergence measure Chakraborty et al. [35] introduced an S-distance measure, to find the dissimilarity between the data points. They have developed an s-k-means algorithm, where the s-distance is used instead of the traditional Euclidean distance measure. The results show that the performance of the s-k-means algorithm is better than the existing approaches especially when the clusters are not regular.

The machine learning technique Conditional Random Fields (CRFs) is used to implement the proposed work. The CRFs is the probabilistic approach which is well suited for

sequential labeling task and it chooses the label sequence y which maximizes the conditional probability of p(y|x) to the observation sequence x [36, 37]. The CRF++ tool kit is used for this work. The features are represented in the template file are learned by the system and the language model for named entity identification is build using the feature template. The CRFs algorithm is explained in section III and the features used for this work are explained in section IV. The various experiments implemented for this work are discussed in section V. The paper concludes in section VI.

A. How the present work differs?

The main objective of the present work and how it differs from the existing works are as follows.

1. Generic NE systems are in great demand for Information Extraction and other NLP applications. The main novelty of the paper is the generic feature selection criteria and applied those features to 12 languages belongs to different language families. The present work focused on the development of generic system to identify named entities irrespective of language and domain.
2. The generic features are extracted automatically from the given dataset. There are no external resources or language specific features used for the system development.
3. The generic features selection methodology used in the proposed work is based on the contextual information and linguistic patterns surrounding the named entities.
4. In order to prove the domain independent property of the proposed work, the dataset used in this work are obtained from various sources like news articles, blogs, newspapers, Wikipedia irrespective of the specific domain.
5. In order to prove the language independent characteristic of the proposed work, first we have applied the generic features to Indian languages such as Tamil, Telugu, Malayalam, Hindi, Bengali, Punjabi, Marathi and then we applied the same features to European languages such as English, Spanish, Dutch, German and Hungarian.
6. For Indian Languages we have used 106 fine grained NE tagset. It is hard to identify named entities in the dataset which consists of hierarchical classes, since it cause more ambiguities in NE classification. Though the dataset used for Indian Languages consists of 106 tags, we have attained encouraging results.

The present work obtained state-of-art performance for all the 12 languages without using any external resources such as gazetteers and language specific features.

II. Corpus Description

A. Diverse Language Families

In order to prove the language independent characteristic of the system, twelve languages which belongs to diverse language families are used in this work. The Indian Languages are mainly classified into two major categories; i) Indo-Aryan ii) Dravidian Language families. Indo Aryan is a subordinate branch of Indo-Iranian language family. The languages like Hindi, Bengali, Marathi, Punjabi which are spoken in North part of India comes under Indo-Aryan Language family. The languages like Tamil, Telugu, Malayalam and Kannda mostly spoken in southern part of India comes under Dravidian Language family. The European language such as English, German and Dutch comes under Germanic language family, Hungarian constitutes a Uralic language family and Spanish belongs to Romance language family.

TABLE I. CORPUS DESCRIPTION

Languages	Tokens	Sentences	NEs
Bengali	52,024	4,030	2,690
Hindi	1,90,236	14,098	14,420
Malayalam	64,345	5,107	11,380
Marathi	73,523	6,138	6,036
Punjabi	85,808	4,714	6,604
Tamil	2,04,144	13,571	22,686
Telugu	43,062	2,150	9,104
English	2,56,426	14,002	25,851
Dutch	2,47,820	15,316	27,390
Hungarian	4,44,661	27,673	7,068
German	5,91,005	31,298	33,399
Spanish	3,17,637	10,238	23,148

B. Indian Language Dataset

The dataset used for Indian Languages and English are obtained from FIRE 2013 NER shared task [28]. The corpus utilized for IL consists of 106 hierarchical Named Entity tags and this is the standard tagset used by the Indian Language Research Community. The NE tags are mainly divided into three categories namely Entity expressions, Numerical Expressions and Time expressions. The Entity expressions are classified into 11 types such as Person, Location, Organization, Facilities, Cuisines, Locomotives, Artifact, Entertainment, Organisms, Plants and Diseases. Numerical expressions are classified into 4 types namely money, distance, quantity and count. Time expressions are classified into 7 types; they are time, period, year, month, date, day and special day. The corpus statistics for each language are shown in Table I. There are 52,024 tokens, 4,030 sentences and 2,690 named entities available in the Bengali corpus. The Hindi corpus consists of 1, 90,236 word forms, 14,098 sentences and 14,420 named entities. The Malayalam, Marathi, Punjabi, Tamil and Telugu dataset consists of 11,380, 6,036, 6,604, and 22,686 named entities respectively.

C. European Language Dataset

The CONLL 2003 NER shared task dataset was utilized for Spanish and Dutch [4]. It consists of 4 NE tags namely person, location, organization and miscellaneous NE tags. The GERMEVAL NER shared task data was used for German language. The 12 NE tags available in the German data which belongs to 4 main NE types namely Person, Location, and Organization, Others [12]. We have used the main class tags for this work. The corpus utilized for Hungarian has been auto generated from the Hungarian Wikipedia pages and it consists of 3 NE tags namely person, location and organization. The English and Dutch NE corpus consists of 2L tokens. The Hungarian, German and Spanish dataset has 4L, 5L and 3L tokens respectively. The numbers of named entities in English and Dutch dataset are 25,851 and 27,390. The Hungarian, German and Spanish

dataset has 7k, 33k, and 23k named entities respectively.

III. Our Methodology

Conditional random fields (CRFs) are a probabilistic framework which is suitable for sequence prediction problem. It selects the label sequence y which maximizes the conditional probability of p(y|x) to the observation sequence x. In this work, the CRF++ toolkit was used for building the language model. The structure of named entities occurred in the training data are learned by the system using the feature template. The language model is automatically generated using the features mentioned in the template file and named entities are identified in the test data.

The algorithm for CRF implementation is given below and the algorithm for the feature selection subroutine computefeature (T,ft) is given in the section IV.

Algorithm 1: CRF Implementation

Input: Training data T and the feature template file ft are the input

Output: NE identified data

1. Define T be the training dataset
2. Let Ck be the number of columns in the training data T, for each k ∈ to 1…m, where m be the number of features
3. Each column Ck in the training data T represents the feature fj, for each j ∈ 1…m
4. Last column in the training data T be the label Yi to be identified by the system, for each i belongs to 1…n, where n be the total number of tokens.
5. To define the features fj in the feature template file, computefeature (T,ft)
6. Then compute the probability of a labeling sequence Y given an observation sequence x using the following formula.

$$P(y \mid x, \lambda) = \frac{1}{z(x)} \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(y_{i-1}, y_i, x, i) \quad (1)$$

$$z(x) = \sum_{i=1}^{n} \sum_{j} \lambda_j f_j(y_{i-1}, y_i, x, i) \quad (2)$$

7. Assign the NE tag or label Y_i to the data sequence X_i if the weight λ_j associated with the feature f_j is large and positive.

Where in (1) and (2) x is the data sequence to be labelled and y is the label sequence. For example x is the range over sentences and y is the range over named entity tag, z is normalization factor, fj (Y_{i-1}, Y_i, x, i) is a state transition feature function of an observation sequence and the labels at position i and i−1. For example, our objective is to assigning the named entity tag or label y "LOCATION" to the sentence x "He went to New Zealand", then the transition function $f_j(y_i\text{-}1, y_i, x, i)=1$ if y_i="LOCATION" and the suffix of i^{th} word is "land"; otherwise 0; If the weight λ_j associated with the above feature is large and positive, then the words ending with the suffix "land" are labelled as NE type "LOCATION" [36, 37].

IV. Feature Selection

Feature selection plays a major role in machine learning. The main novelty of the proposed work is that an end user can use the system without deep knowledge about the language or domain. Hence the features used are generic and not depend on any external or language specific resources. The features used for the system development are as follows

A. Lexical level Features

The features based on words or tokens are considered as lexical level features.

1) Context word information

The context information denotes the tokens occurring adjacent to the named entity. The contextual words will be useful to identify an NE. Hence, the combinations of words surrounding the current token are considered as features.

2) First word

We have analyzed the training corpus for all the languages, to check the occurrences of named entities which occurred as first word of the sentence. Since subject comes in the beginning of the sentence, the subject might be a named entity in most of the languages. We observed that around 20% of the words occurring at beginning of a sentence in the training corpus are NEs for all the languages. So we consider the beginning word of the sentence as a feature.

3) Prefix and Suffix Information

In all the languages, Named entities can share common prefixes and suffixes. For example, the words ending with "land" most likely specifies the place names such as "Finland, England, New Zealand and Ireland" in English languages. Hence we consider trigrams and four grams of prefix and suffix information as features.

B. Syntactic level Features

The features depends on the linguistic or grammatical information are called as syntactic level features

1) PoS information

PoS tags denote the grammatical category of the word. It is considered as a crucial factor in NE identification. The PoS information of a current word and surrounding words are considered as feature.

C. Dynamic Features using Frequency based approach

Dynamic features are the features which are automatically extracted from the given dataset by using the *frequency based approach*. The weightage or threshold given to the dynamic features are based the frequency of the occurrences of the respective features in the corpus.

1) PoS patterns preceding & following NE

The PoS tags occurring as context of named entity act as a trigger for NE identification. Hence we have decided to give importance to the PoS tags preceding & succeeding named entities.

2) Preceding word and PoS information of NE

The words surrounding the named entities act as an indicator in NE identification. For example, the proper nouns following by designations can be a person name. The words preceding the named entity and the PoS tag of NE in the training corpus are extracted based on the feature selection procedure mentioned above.

3) Following word and PoS information of NE

The words following named entities are a trigger in some instances. There are some key words which followed by proper nouns can be useful to predict named entities. For example, the verbs such as "play, read, lived, located, situated" follow or proceeded by person or location names.

Hence context words occurred in the proximity of NE in w_{i+1} position and the PoS tag of NE can be extracted from training data using the selection procedure.

Algorithm 2: Feature Selection

Input: Training data T and feature template ft is the input

Output: Template file with features

Procedure computefeature (T,ft)

1. Let T be the training data
2. Consider the words w_i, w_{i+1},w_{i-1} of sentence S_i in the moving window of 3 as a feature
3. Identify the first word fw_i of sentence S_i as a feature
4. Identify the trigram and 4-gram characters from the end of the words w_i, w_{i+1},w_{i-1}.
5. Take the trigram and 4-gram characters from the beginning of the words w_i, w_{i+1},w_{i-1}.
6. Consider the Part of the Speech(POS) tag, POS_i for each token
7. Find the frequent POS patterns using freqbasedapproach (T, pt_i), where pt is the pattern respective to the word in the ith position.
8. Find the frequent word and POS patterns preceding the NE using freqbasedapproach (T, pt_i), where pt is the pattern respective to the word in the ith position.
9. Find the frequent word and POS patterns following the NE using freqbasedapproach (T, pt_i), where pt is the pattern respective to the word in the ith position.

End Procedure

Algorithm 3: Frequency Based Approach

Input: Training data T and patterns pt is the input

Output: Frequent patterns

Procedure freqbasedapproach(T, pti)

1. Compute the pattern weight (PATTERNweight) for each pattern pt_i using the given formula

$$PATTERN_{weight} = \sum_{i=1}^{n} \frac{NEP_i}{TP} \quad (3)$$

2. Divide the patterns into 3 partitions (sets) by calculating the average frequency range of the pattern ($AVG_{freqRANGE}$)

$$AVG_{freqRANGE} = \frac{TUP}{3} \quad (4)$$

3. Find the frequency position fp of the pattern pt_i from the list of patterns and assign the pattern pt_i to one of the 3 sets namely, $MOST_{freqSET}$, $SECOND_{freqSET}$ and $LEAST_{freqSET}$

   If frequency position fq of the pattern pt_i ($fq \le AVG_{freqRANGE}$)

   Then the pattern pt_i comes under $MOST_{freqSET}$

   If frequency position of the pattern (($fq \ge AVG_{freqRANGE}$)&&($fq \le AVG_{freqRANGE}*2$))

   Then the pattern pt_i comes under $SECOND_{freqSET}$

   If frequency position of the pattern (($fq \ge AVG_{freqRANGE}*2$)&&($fq \le TUP$))

   Then the pattern pt_i comes under $LEAST_{freqSET}$

4. Assign the threshold th for each pattern pt_i and remove the patterns that are failed to meet the criteria.

   If the pattern pt_i comes under $MOST_{freqSET}$ and minimum $PATTERN_{weight}$ is >= 0.60
   then the pattern pt_i is selected
   If the pattern pt_i comes under $SECOND_{freqSET}$ and minimum $PATTERN_{weight}$ is >= 0.75
   then the pattern pt_i is selected
   If the pattern pt_i comes under $LEAST_{freqSET}$ and minimum $PATTERN_{weight}$ is >= 0.80
   then the pattern pt_i is selected

Note: Where in (3) NEP – occurrences of the pattern occurred as part of NE, TP – total occurrences of the pattern in the corpus, where in (4) TUP – total number of unique patterns in the corpus.

D. Validation using K-means clustering

The dynamic features discussed above are selected based on the frequency based approach which is explained in algorithm 3. In frequency based approach a weightage is given to the patterns based on the frequency of the occurrences of the respective patterns. In order to validate the frequency based approach we have used K-means clustering algorithm using WEKA datamining tool. The lists of patterns are given as the input to the k-means algorithm, the frequent patterns from the input data forms the cluster heads. From the experiments we have observed that the patterns selected using the frequency based approach forms the cluster head in K-means++ algorithm. The figure 1 shows the POS patterns preceding the named entities in the input data, for example NN denotes the noun tag which precedes the named entities 5,754 times. From figure 2 it is clearly shown that the NN – noun tag has the higher density of 42% in the corpus, which is similar to the patterns we obtained using the frequency based approach. Thus we have validated the frequency based approach using K-means++ algorithm.

Algorithm 4: Feature selection using K-means++

Input: P patterns in the training data; K denotes the total number of clusters

Output: K centroids {c1, c2, … ck}

1. Extract the patterns P from the training data
2. Define the number of clusters k
3. Choose the initial cluster head c_i where $c_i = p' \in P$ with the probability

$$prob = \frac{dist(p'^2)}{\sum_{p \in P} dist(p^2)} \propto dist(p'^2)$$

4. Until k cluster heads for the pattern has been sought
5. Return(c_1, c_2, … c_k)

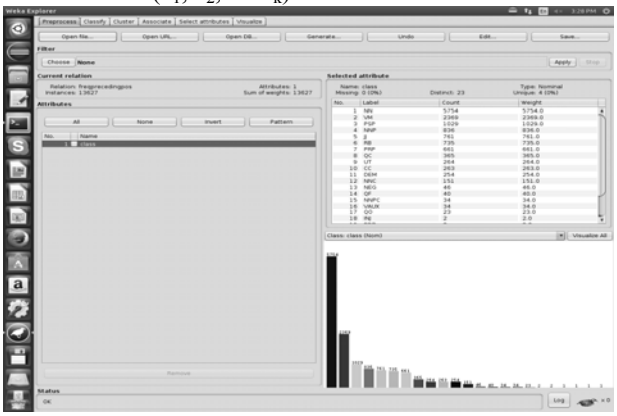

Figure 1. Validation of feature selection using K-means clustering

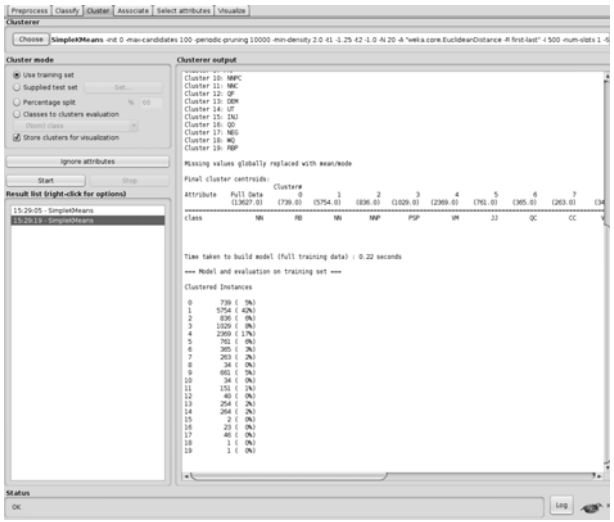

Figure 2. Clustered instances of preceding POS patterns

V. EXPERIMENTS & RESULTS

For these experiemnts, we have used the NER dataset belongs to 7 Indian languages and 5 European languages. We have provided the web demo interface for Named Entity Recognition (NER) for two languages namely Tamil and Dutch in the following link.
http://78.46.86.133:8080/multine-demo/.

The Tamil NER Engine Software for download is given in the following link http://78.46.86.133:8080/multine-demo/TAMIL_NER_ENGINE_SOFTWARE.tar.gz

The Dutch NER Engine Software for download is given in the following link http://78.46.86.133:8080/multine-demo/DUTCH_NER_ENGINE_SOFTWARE.tar.gz

The detail statistics of the dataset used in this work is explained in section II corpus description. The corpus used for Indian Languages and English are obtained from FIRE 2013 NER shared task [28]. The GERMEVAL NER shared task data [12] was used for German and CONLL 2002 NER shared task [4] dataset was used for Spanish and Dutch languages. The structure of the training data used in the experiments are similar to the standard datasets like CONLL NER task. For all the languages we have used the NE labelled dataset. The named entites are tagged in the BIO(Beginning, Inside, Outside) format. The experiments were conducted first for Indian language corpus. We have performed 10-fold cross validation. The dataset was divided into 10 partitions for all the languages. We have considered 9 partitions as a training set and remaining one partition as test set. Language Independent features like word, PoS, bigrams and trigrams of suffixes & prefixes, PoS tags preceding and following NE and words preceding and following NEs are extracted from the training data to build the language model. The language model generated by the CRFs was applied on the test data to identify the named entities. In order to show the language independent characteristic of the proposed work, we have applied the same features to European Languages which belongs to a different language family. We have conducted various experiments with different combinations of features and the average f-score obtained by 10 fold cross-validation for the combination of best feature sets for Dravidian, Indo-Aryan and Indo-European languages are given in Table II, Table III and Table IV respectively.

In the table W-word, POS-part of speech related features, AFF – Affixes (prefix and suffix information), DNF – Dynamic Features.

TABLE II. FEATURE WISE F-MEASURE FOR DRAVIDIAN LANGUAGES

Languages	W + POS	W+ POS+ AFF	W+ POS + AFF + FW	W+ POS + AFF + FW + DNF
Tamil	64.12	67.32	69.34	76.60
Telugu	51.31	53.25	54.13	59.80
Malayalam	59.54	61.15	62.32	65.83

The baseline system was developed using context word and PoS information. Among Dravidian languages, Tamil has achieved F-score of 64% for the baseline system. Using word and PoS features Telugu & Malayalam languages has attained F-score of 51.31% and 59.54% respectively. Since named entities have some specific prefixes and suffixes, we have used the trigrams and bigrams of prefix and suffix information of the current token as a feature. The inclusion of suffix & prefix features improves the f-score by 3.2% for Tamil, 1.94% for Telugu and 1.67% for Malayalam. In most of the languages subject might be a named entity and probably it occurs at the beginning of the sentence. Therefore we considered the first word as one of the feature; the results also show that addition of first word features further improves results by 1% to 2% for all the languages. The fifth column in table 2 reveals that inclusion of dynamic features increases the f-score by 7.26% for Tamil, 5.67% for Malayalam and 3.51% for Telugu respectively.

TABLE III. FEATURE WISE F-MEASURE FOR INDO-ARYAN LANGUAGES

Languages	W + POS	W + POS + AFF	W + POS + AFF + FW	W + POS + AFF + FW + DNF
Bengali	67.18	71.21	72.86	78.12
Hindi	68.20	70.50	72.16	77.69
Punjabi	67.43	71.18	72.69	76.40
Marathi	66.10	72.10	72.62	77.74

We observed that the baseline system obtained F-score of above 65% for all the Indo-Aryan languages. Usage of suffix & prefix information increases the f-score by 4.03%, 2.30%, 3.75% and 4.10% for Bengali, Hindi, Punjabi and Marathi respectively. Evaluation results show that the addition of first word feature improves the f-score by approximately 2% for all the languages mentioned in Table III. Results shown in fifth column indicates that the inclusion of dynamic features has a significant role in the improvement of system's performance, so that the f-score further increased by 4%-5% for the Indo-Aryan languages.

TABLE IV. FEATURE WISE F-MEASURE FOR EUROPEAN LANGUAGES

Languages	W + POS	W + POS + AFF	W + POS + AFF + FW	W + POS + AFF+ FW + DNF
English	70.19	74.23	76.11	82.29

Spanish	71.32	75.34	77.32	85.24
Dutch	75.45	81.32	83.32	94.21
German	66.78	70.34	71.23	76.90
Hungarian	62.65	67.32	68.22	73.05

The evaluation scores in second column of Table IV show that the use of context word and PoS information yields reasonable results for the base line system. The suffix information improved the system's performance by 4%-5% (F-score) for the European languages. The usage of first word feature also improves the f-score by 1%-2% for all the languages mentioned in Table IV. Results in column (5) shows that due to the addition of dynamic features performance of the system was further improved by 4% to 7% for all the European languages.

The results in Table V clearly show that all the features used in the present system have the capability to improve the system's performance. The results show that our feature selection strategy yields reasonable results not only for Indian Languages but also for European languages. By using only context word and PoS information we have achieved reasonable scores for baseline system. From the results we can clearly understand that the dynamic features highest improvement is obtained by using the dynamic features. In general, the usage of suffix, prefix information and inclusion of dynamic features such as word, PoS, bigrams and trigrams of suffixes & prefixes and PoS tags preceding and following NE, words preceding and following NE significantly improves the overall performance of the system for all the languages. From the results, it is observed that the usage of affix information increases the f-score by 1.67%-4% and due to the effectiveness of dynamic features; overall performance has been improved by 3%-7% for all the languages.

The feature wise improvement for Dravidian, Indo-Aryan and European languages are shown in figure 3, 4 and 5 respectively. The results for Indian & European languages in terms of precision, recall and f-score is given in Table V and Figure 6. The number of NE tags used in existing works is less than the proposed work where 106 NE tags used for Indian languages. Without using language independent features and gazetteers we achieved state-of-art performance for all the languages. The results obtained for European languages like Spanish, Dutch and English are higher than the Indian languages. The performance fall in Indian Languages are mainly due to its rich morphology, agglutinative nature, spell variations and free word order structure. Indian Languages belongs to the Indo-Aryan and Dravidian language families. Hindi, Punjabi, Bengali and Marathi belong to Indo-Aryan and south Indian languages like Tamil, Telugu & Malayalam come under Dravidian family. Like English, European languages have subject, object, and verb sentence structure. But, Indian languages have free word order. The agglutination and inflections results in long and complex word forms which further makes the NE identification difficult for Indian Languages.

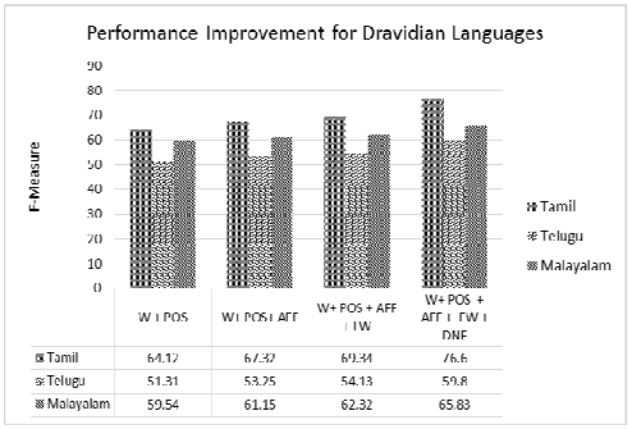

Figure 3. Feature wise performance improvement for Dravidian languages

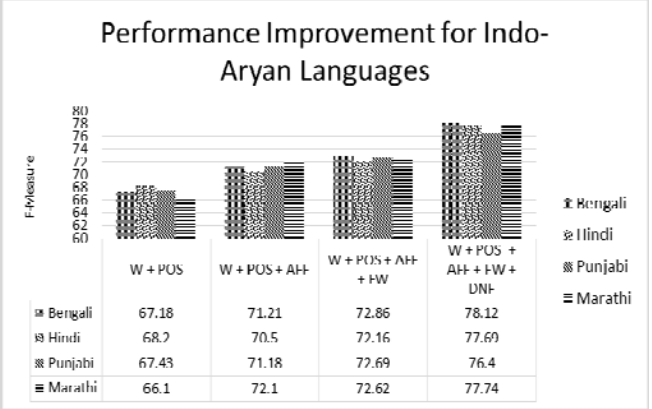

Figure 4. Feature wise performance improvement for Indo-Aryan languages

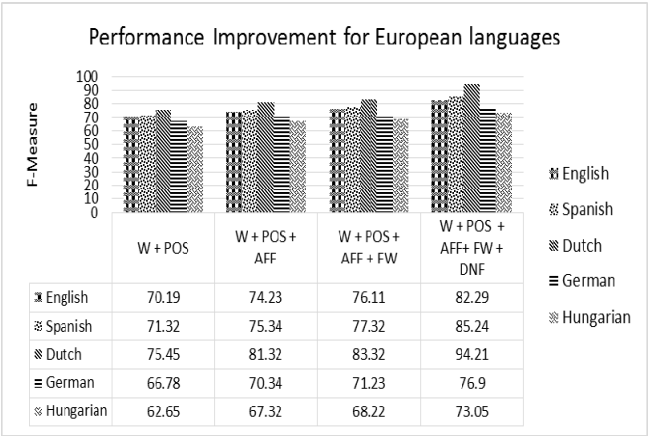

Figure 5. Feature wise performance improvement for European languages

TABLE V. RESULTS FOR INDIAN AND EUROPEAN LANGUAGES

Languages	Precision	Recall	F-Measure
Tamil	80.12	73.36	76.6
Hindi	81.05	74.59	77.69
Malayalam	70.63	61.64	65.83
Marathi	81.32	74.16	77.74
Punjabi	80.54	72.27	76.4
Bengali	82.78	73.46	78.12
Telugu	69.4	50.21	59.8
English	84.32	80.35	82.29
Spanish	86.13	84.37	85.24
Dutch	93.3	95.12	94.21

German	81.41	72.99	76.9
Hungarian	93.84	59.8	73.05

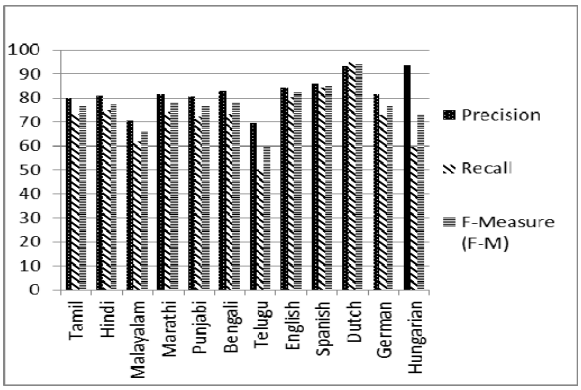

Figure 6. Overall results for Indian and European Languages

A. Comparison with existing Systems

In this section, the performance of the present work is compared with the existing works already discussed in Introduction section. Since the present work is about multilingual NER, in this section we have compared the performance of the present work with the previous works related to multilingual NE systems. Gayen et al. [27] has participated in ICON NER shared task and developed a NE system for English and Indian languages such as Bengali, Hindi, Marathi, Punjabi Tamil and Telugu. The dataset consists of 22 NE tags. They have used generic features such as word, PoS, chunk, suffix & prefix information for NE identification using HMM. They have achieved the f-score value of 85.99% for Bengali, 77.04% for English, 75.20% for Hindi, 42.89% for Marathi, 54.55% for Punjabi, 44% for Tamil and 40.03% for Telugu. Even though the present system has used fine grained NE tags for Indian languages, we have obtained highest f-score for all the Indian languages except Bengali in comparison with the results reported by Gayen et al. Abinaya et al. [30] has participated in FIRE 2014 shared task and built a NE system for English, Hindi, Tamil and Malayalam. The FIRE 2014 corpus consists of 106 NE tags and nested entities. The results given in 4th column of 2nd row in Table VII, are obtained by Abinaya et al. for maximal entities as reported in FIRE 2014 NER task overview paper [29]. They have used CRFs for English and SVM for other languages. The features applied are word, PoS, chunk affixes, digit features, word length and binary features. For Malayalam and Tamil language gazetteer has been used in their work. From the results reported by Abinaya et al. we observe that, it is challenging to develop the NE system with fine grain tagset, especially when we focus on multiple languages. Without using gazetteer or other external resources we have obtained better results than their work. The language Independent (LI) NE system has been developed for Hindi and Bengali using CRFs by Ekbal et.al [18]. The main features used in their work were word, PoS, previous NE tag, first word, orthographic and digit features. The results attained by the present work in Bengali language are 0.38% greater and Hindi is 0.61% less than [18].

Florian et al. [11] participated in CONLL 2002 NER shared task and obtained 79.05% for Spanish and 74.99% for Dutch. The features used were word, PoS, chunk, capitalization and dictionary. In comparison with their work, the present work got 6.19% highest score for Spanish and 19.22% better score for Dutch. Benikova et al. [12] obtained best f-score value of 77.14% in GermEval shared NER task using machine learning technique CRFs. The features used were word, PoS, chunk, word shape features and similarity clusters. The f-score secured by the present work is 0.24% lesser than the results reported by [12]. Szarvas et al. [10] used voting based approach for Hungarian NER and scored f-score value of 91.95%. They have used word, PoS, word length and gazetteer information as features for system development. We have achieved 73.05% for Hungarian corpus which is lesser than their work.

TABLE VI. COMPARISON WITH EXISTING SYSTEMS (METHODS AND CORPUS DETAILS)

Existing Systems	**Methods**	**Corpus & NE-Tags**
Gayen et al. [27]	HMM	ICON NER Shared task corpus NE-Tags: 22
Abinaya et al. [30]	CRF for English SVM for other languages	FIRE 2014 NER shared Task Corpus NE Tags:106
Ekbal et al. [18]	CRF	IJCNLP-08 NER shared task corpus NE Tags: 12
Florian et al. [11]	Stacking based approach (TBL, SNOW, Forward backward algorithm)	CONLL-2002 NER shared task corpus NE Tags: 4
Our system	CRFs	FIRE 2013 NER shared task corpus for English and Indian Languages with 106 NE tags, CONLL-2002 NER shared task corpus for Spanish & Dutch, GermEval NER shared task data for German, Wikipedia corpus for Hungarian

TABLE VII. COMPARISON WITH EXISTING SYSTEMS (RESULTS)

Existing Systems	**Features/ Resources**	**Languages used**	**F-M**
Gayen et al. [27]	Word, PoS, chunk, suffix information	Bengali English Hindi Marathi Punjabi Tamil Telugu	85.99 77.04 75.20 42.89 54.55 44.00 40.03
Abinaya et al. [30]	Word, PoS, chunk, affixes, digit features, word length, binary features and gazetteers for Tamil & Malayalam	English Hindi Tamil Malayalam	57.81 25.53 30.75 24.91
Ekbal et al. [18]	Word, PoS, previous NE tag, first word, digit features, orthographic	Bengali (LI) Hindi (LI)	77.74 77.08

	features, word length, infrequent word(for LI)		
Florian et al. [11]	Word, PoS, chunk, capitalization, dictionary	Spanish Dutch	79.05 74.99
Our system	Word, PoS, first word, trigrams & bigrams of suffixes & prefixes of current token, PoS patterns preceding & following NE, Preceding word and PoS information of NE, Following word and PoS information of NE	Bengali Hindi Marathi Punjabi Tamil Telugu Malayalam English Spanish Dutch German Hungarian	78.12 77.69 77.74 76.40 76.60 59.80 65.83 82.29 85.24 94.21 76.90 73.05

B. Error Analysis

We have analysed the results where the system failed to identify the NEs. From our observation, mainly the errors are propagated due to the following reasons.

1) Ambiguity due to Nested Entities

Ambiguity arises when one type of entity occurring within another entity (nested entities). If an entity is considered as maximal entity, it belongs to one NE type and the entities occurred as part of it comes under different NE category. In such cases, the system might fail to identify the whole entity and mark only the part of that which comes under another NE category

Example

Ta: *SrIrafkam peVrumAlY* mAwiri ifkum
En: Srirangam(N) perumal(N) like(N) here(ADV)
Ta: varatharaja(N) *peVrumAlY* weVrYku nokki
En:varatharaja perumal(N) south(N) towards(ADV)
Ta: kAtci alYikkirYAr
En: bestowing(V)+present+3s
(Like, Srirangam Perumal, here also Varatharaja Perumal bestowing towards south)

In this example, instead of tagging the maximal entity *SrIrafkam peVrumAlY* system tagged the entity *SrIrafkam* as PLACE and *peVrumAlY* as PERSON.

2) Ambiguity between common and proper nouns

In comparison with European languages, Indian Languages have more ambiguity between proper and common nouns. In our corpus, we observed that some common nouns found in the dictionary are occurring also as person names which create ambiguity issues.

Example

Ta: iwarYkAka *cakwi*, pUjE ceVAlY
En: For this(ADV) sakthi(N) pooja(N) do(V)+past+3sf
(For this sakthi did pooja)

Example

Ta: pakEvarkalYE veVllum cakwiyEyum
En: Enemies(N)+acc win(V)+future power(N)+acc
Ta: waruvawAl iwwalawwu
En: give(V)+future+cond this place(N)+gen
Ta: ammanY mikavum AkrocamAka *cakwi*
En: Goddess(N) more(ADV) aggressive(N) power(N)
Ta: ulYlYavar.
En: has(v)+PRP
(The goddess of this place is very aggressive, as she gives power to win the enemies)

In example2 "*cakwi(sakthi)*" is a person who did pooja to God. In second example *cakwi* is not a person name, it means "power". Here in first example "*cakwi*" is a named entity, but in second example it is a common noun. When the same word occur as named entity and common noun ambiguity issue will occur.

3) Genitive drop in Sequential NEs

The genitive case marker indicates the possessive form. The genitive drop in sequential named entities creates an ambiguity in some instances

Example

Mangeshkar has sung for almost all the *Yash Raj* films and films from his production house *Yash Raj Films*

For example, In the above sentence, consider the phrase "*Yash Raj films*", where "*Yash Raj*" is a person name and Mangeshkar sung for most of his films. Here the absence of possessive marker "'s" creates a ambiguity whether "*Yash Raj films*" is a named entity or the entity "*Yash Raj*" is a named entity. But the phrase at the end of the sentence "*Yash Raj Films*" is not a person name, it is the name of the production house established by *Yash Raj*. Due to ambiguity, system wrongly tagged the organization name as person; in this case system recognized "*Yash Raj*" as PERSON category which is a part of an entity "*Yash Raj Films*".

Example

Ta: *Chennai merinA kadaRkarai* Aciyavin
En: Chennai(N) marina(N) beach(N) Asia(N)+gen
Ta: irantavathu neelamana kadaRkarai Akum
En: second(QT) longest(ADV) beach(N) is(V)

In the above example, absence of genitive case marker "*-in"* causes the ambiguity whether the entity *Chennai merinA kadaRkarai* as a whole is a NE or *Chennai and merinA kadaRkarai* are separate NEs**.** Actually in this example *Chennai and merinA kadaRkarai* are different NEs where *merinA kadaRkarai* is a beach in *Chennai***.**

The errors discussed in the error analysis can be resolve using post-processing rules and incorporating dictionaries for location and popular organization names for the respective languages. For example, if the keyterm such as "Temple" following the proper noun, then the "PERSON" should change to "LOCATION" tag. Since our intention is to build the generic NER engine across languages, no post-processing rules or gazetteer lists are used in this work.

VI. CONCLUSION

We have presented the language & domain independent, crosslingual named entity system which can be identifying named entities for any languages without deep knowledge about the language. The generic features are extracted automatically from the given dataset. In order to show the language independence characteristic of our system, the generic feature selection methodology was first applied to 7 Indian Languages and then for 5 European Languages. The results obtained show that our generic features are well suited to identify named entities across languages with diverse language families and sentence structures. There are no external resources or language specific features used for the system development. In future, we plan to extend this

work towards the development of the generic information extraction system.

REFERENCES

[1] A. Borthwick, J. Sterling, E. Agichtein, R. Grishman, "NYU: Description of the MENE named Entity System," in Proc. Seventh Machine Understanding Conference (MUC-7), Virginia, 1998.

[2] D. Nadeau, S. Sekine, "A survey of named entity recognition and classification," Linguisticae Investigationes, vol. 30, no. 7, pp. 3–26, 2007. doi: 10.1075/li.30.1.03nad

[3] D.M. Bikel, S. Miller, R. Schwartz, R. Weischedel, "Nymble: A high-performance learning name-finder," in Proc. Fifth Conference on Applied Natural Language Processing, Washington, 1997, pp. 194-201. doi:10.3115/974557.974586

[4] E.F. Tjong Kim Sang, "Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition," in Proc. CONLL-2002, Taipei, Taiwan, 2002, doi:10.3115/1118853.1118877

[5] E.F. Tjong Kim Sang, F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in Proc. of the seventh conference on Natural language learning at HLT-NAACL 2003, Canada, vol. 4, 2003, pp. 142-147. Arxiv:cs/0306050

[6] R. Florian, A. Ittycheriah, H. Jing, T. Zhang, "Named entity recognition through classifier combination," in Proc. Seventh conference on Natural language learning at HLT-NAACL 2003, ACM, vol. 4, pp. 168-171, 2003. doi:10.3115/1119176.1119201

[7] F. De Meulder, W. Daelemans, "Memory-based named entity recognition using unannotated data," in Proc. Seventh conference on Natural language learning at HLT-NAACL 2003, ACL, vol. 4, 2003, pp. 208-211. doi:10.3115/1119176.1119211

[8] B. Desmet, V. Hoste, "Dutch named entity recognition using classifier ensembles," LOT Occasional Series, vol. 16, pp. 29-41, 2010.

[9] D. Varga, E. Simon "Hungarian named entity recognition with a maximum entropy approach," Acta Cybern, vol. 18, no. 2, pp. 293-301, 2007.

[10] G. Szarvas, R. Farkas, A. Kocsor, "A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms," in Proc. International Conference on Discovery Science, pp. 267-278, 2006. doi:10.1007/11893318_27

[11] R. Florian, "Named entity recognition as a house of cards: Classifier stacking," in Proc. of the 6th conference on Natural language learning, Association for Computational Linguistics, vol. 20, pp. 1-4, 2002. doi:10.3115/1118853.1118863

[12] D. Benikova, C. Biemann, M. Kisselew, S. Padó, "Germeval 2014 named entity recognition shared task: companion paper," in Proc. KONVENS GermEval Shared Task on Named Entity Recognition, Hildesheim, Germany, 2014, pp. 104-112.

[13] A.K. Singh, "Named Entity Recognition for South and South East Asian Languages: Taking Stock", in Proc. IJCNLP, India, 2008, pp. 5-16.

[14] S. K. Saha, P. Sarathi Ghosh, S. Sarkar, P. Mitra, "Named entity recognition in Hindi using maximum entropy and transliteration," Polibits, vol. 38, pp. 33-41, 2008. doi: 10.17562/PB-38-4

[15] S. Gupta, P. Bhattacharyya, "Think globally, apply locally: using distributional characteristics for Hindi named entity identification," in Proc. Named Entities Workshop, 2010, pp. 116-125. ISBN: 978-1-932432-78-7

[16] N.V. Patil, A.S. Patil, B.V. Pawar, "Issues and Challenges in Marathi Named Entity Recognition," International Journal on Natural Language Computing (IJNLC), vol. 5, no. 1, pp. 15-30, 2016. doi: 10.5121/ijnlc.2016.5102

[17] A. Kaur, G.S. Josan, "Evaluation of Named Entity Features for Punjabi Language," Procedia Computer Science," vol. 1, no. 46, pp. 159-166, 2015. doi: 10.1016/j.procs.2015.02.007

[18] A. Ekbal, S. Bandyopadhyay, "A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi," Linguistic Issues in Language Technology, vol. 2, no. 1, pp. 1-44, 2009.

[19] K.N. Kumar, G.S.K. Santosh, V. Varma, "A Language-Independent Approach to Identify the Named Entities in under-resourced languages and Clustering Multilingual Documents," in Proc. International Conference on Multilingual and Multimodal Information Access Evaluation, Amsterdam, 2011, pp. 74-82.

[20] M.S. Bindu, I. Sumam Mary, "Design And Development Of A Named Entity Based Question Answering System For Malayalam Language," PhD diss., Cochin University Of Science And Technology, 2012.

[21] G.V.S. Raju, B. Srinivasu, S.V. Raju, K.S.M.V. Kumar, Named Entity Recognition for Telugu using Maximum Entropy Model. Journal of Theoretical & Applied Information Technology, vol. 1, no. 13, 2010.

[22] S. L. Pandian, T.V. Geetha, Krishna, "Named Entity Recognition in Tamil using Context-cues and the E-M algorithm," in Proc. 3rd Indian International Conference on Artificial Intelligence, Pune, India, pp. 1951-1958, 2007. doi: 10.1109/IALP.2009.26

[23] R. Vijayakrishna, L.D. Sobha, "Domain focused Named Entity for Tamil using Conditional Random Fields," in Proc. workshop on NER for South and South East Asian Languages, Hyderabad, India, 2008, pp. 59-66.

[24] C.S. Malarkodi, L.D. Sobha, "A Deeper Look into Features for NE Resolution in Indian Languages," in Proc. of the Workshop on Indian Language Data: Resources and Evaluation, LREC, Istanbul, 2012, pp. 36-41.

[25] C.S. Malarkodi, R.K. Pattabhi, L.D. Sobha, "Tamil NER–Coping with Real Time Challenges," in Proc. workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012), COLING, Bombay, India, 2012, pp. 23-38.

[26] L.D. Sobha, C.S. Malarkodi, K. Marimuthu, "Named Entity Recognizer for Indian Languages," in Proc. ICON NLP Tool Contest, India, 2013.

[27] V. Gayen, K. Sarkar, "An HMM based named entity recognition system for Indian languages: the JU system at ICON 2013," in Proc. of the ICON NLP Tool Contest, 2014. arXiv:1405.7397v1

[28] R.K. Pattabhi, L.D. Sobha, "NERIL: Named Entity Recognition for Indian Languages @ FIRE 2013–An Overview," in Proc. FIRE-2013, India, 2013.

[29] R.K. Pattabhi, L.D. Sobha, "NERIL: Named Entity Recognition for Indian Languages @ FIRE 2014–An Overview," in Proc. of the FIRE-2014, India, 2014.

[30] N. Abinaya, J. Neethu, H.B.G. Barathi, M.K. Anand, K.P. Soman, "AMRITA_CEN@ FIRE-2014: Named Entity Recognition for Indian Languages using Rich Features," in Proc. Forum for Information Retrieval Evaluation, India, ACM, 2014, pp. 103-111. doi:10.1145/2824864.2824882

[31] S.K. Saha, S. Sudeshna M. Pabitra, "Feature selection techniques for maximum entropy based biomedical named entity recognition," Journal of biomedical informatics, vol. 42, no. 5, pp. 905-911, 2009. doi:10.1016/j.jbi.2008.12.012

[32] S. Zahra, M.A. Ghazanfar, A. Khalid, M.A. Azam, U. Naeem, & A. Prugel-Bennett, "Novel centroid selection approaches for KMeans-clustering based recommender systems," Information sciences, vol. 320, pp. 156-189, 2015. doi:10.1016/j.ins.2015.03.062

[33] T. Zhang, F. Ma, "Improved rough k-means clustering algorithm based on weighted distance measure with Gaussian function," International Journal of Computer Mathematics, vol. 94, no. 4, pp. 663-675, 2017. doi:10.1080/00207160.2015.1124099

[34] I.D. Borlea, R.E. Precup, F. Dragan, A.B. Borlea, A. B. "Centroid update approach to K-means clustering," Advances in Electrical and Computer Engineering, vol. 17, no. 4, pp. 3-11, 2017. doi: 10.4316/AECE

[35] Chakraborty, Saptarshi, D. Swagatam, "k− Means clustering with a new divergence-based distance metric: Convergence and performance analysis," Pattern Recognition Letters, vol. 100, pp. 67-73, 2017. doi:10.1016/j.patrec.2017.09.025

[36] J. Lafferty, A. McCallum, F. Pereira, "Conditional Random Fields for segmenting and labelling sequence data," in Proc. ICML-01, Massachusetts, 2001, pp. 282-289.

[37] H.M. Wallach, "Conditional random fields: An introduction," Technical Reports (CIS), MSCIS-04-21, 2004.