ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(24), DOI: 10.17485/ijst/2015/v8i24/85350, September 2015* ISSN (Online) : 0974-5645

Randomized Kernel Approach for Named Entity Recognition in Tamil

N. Abinaya*, M. Anand Kumar and K. P. Soman

Centre for Excellence in Computational Engineering and Networking, Amrita Vishwa Vidyapeetham, Coimbatore - 641 112, Tamil Nadu, India;
abi9106@gmail.com, m_anandkumar@cb.amrita.edu, kp_soman@amrita.edu

Abstract

In this paper, we present a new approach for Named Entity Recognition (NER) in Tamil language using Random Kitchen Sink algorithm. Named Entity recognition is the process of identification of Named Entities (NEs) from the text. It involves the identifying and classifying predefined categories such as person, location, organization etc. A lot of work has been done in the field of Named Entity Recognition for English language and Indian languages using various machine learning approaches. In this work, we implement the NER system for Tamil using Random Kitchen Sink algorithm which is a statistical and supervised approach. The NER system is also implemented using Support Vector Machine (SVM) and Conditional Random Field (CRF). The overall performance of the NER system was evaluated as 86.61% for RKS, 81.62% for SVM and 87.21% for CRF. Additional results have been taken in SVM and CRF by increasing the corpus size and the performance are evaluated as 86.06% and 87.20% respectively.

Keywords: Conditional Random Field (CRF), Named Entities (NEs), Named Entity Recognition (NER), Natural Language Processing (NLP), Random Kitchen Sink (RKS), Support Vector Machine (SVM)

1. Introduction

Information Extraction (IE) is the process of extracting knowledge from text. The goal of Information Extraction (IE) systems is to find and understand some degree of relevant parts of the text. It produces some structured representation of related information (e.g.: Relations in the database). The main significant purpose of Information Extraction systems is to organize information and order the information in a semantically precise form that allows further inferences to be made by computer algorithms. It also extract a clear and accurate information (e.g.: who did what to whom when?).

The Named Entity Recognition (NER) is a subtask of Information extraction that mainly seeks to identify and classify entities in text into the predefined classes such as the names of persons, organizations, locations, time and date, measures etc. Depending on the application, NER sub entities increases. Named Entity Recognition is also referred as an object identification and object extraction. It is the sub component of larger task for many of the NLP applications. NER is also applied in sentiment analysis to extract named entities such as names of companies and product names and question answering applications, answers are often named entities.

Generally, NER is implemented by rule based and machine learning techniques. Rule based approach can be language dependent or language independent system which requires expertise in language. The changes are hard to accommodate in rule based approach for language like Tamil. Machine learning systems are learning systems which are independent of language and does not require language engineers. The supervised and unsupervised approach of machine learning system uses statistical methods such as Support Vector Machine (SVM), Conditional Random Field (CRF), Hidden Markov Model (HMM) etc. The trend in Machine Learning is to provide domain knowledge with large data set. Natural Language Processing applications use huge data set for giving knowledge to the system. One such non-linear statistical

* *Author for correspondence*

method used in this paper for classifying named entities in Tamil is Random Kitchen Sink (RKS) algorithm.

Random Sink Algorithm (RKS) is a machine learning algorithm for nonlinearly separated data sets. The main advantage of Random Kitchen Sink algorithm is independent of the number of data points. So, this method is suitable for systems like Natural Language Processing applications which require large dataset. When large nonlinear data set are used for SVM classification, large proportion of data points become support vectors and these support vectors are stored for classifying new data point but RKS considers only the feature vectors for classification task.

In this work, Named Entity Recognition is the classification task done for Tamil using Random Kitchen Sink algorithm which is a new approach in NER. Section 2 describes the works done previously on Named Entity Recognition, Section 3 deals with the mathematical formulation of various machine learning algorithms, the obtained results are discussed in Section 4 and this work is concluded in Section 5.

2. Related Works

Ali Rahimi and Benjamin Recht[2] presented random features which is an economical and powerful tool for supervised machine learning system in 2007. They focused this feature for classification and regression task. It can also be used for other kernel methods like semisupervised and unsupervised models. Again in 2009, Rahimi and Recht[3] selected the random nonlinearities and achieved the accuracy as a greedy algorithm. The nonlinearities will transform the input so as they can be linearly separable and this transformation is provided by random featurization.

In 2005, finite automata acquisition algorithm is used for NER by Muntsa and Lluis[12] for Spanish. This system opens the door for applying the proposed algorithm for various NLP tasks such as POS tagging and chunking. David Nadeau et al.[13] proposed an unsupervised system in 2006 for classifying more than three classical named entities (person, location and organization). Human intervention is not required by this system for labeling training data or creating a gazetteer. The proposed system by Nadeau outperforms the supervised system. A new approach was developed to identify named entities using Phonetic matching technique by Animesh et al.[14] in 2008. In this technique, the strings of the different languages are matched based on the sounding property. It is developed for Hindi and English corpus which is language independent and requires few rules appropriate for specific language.

An integrated machine learning approach between bootstrap-ping which is semisupervised and Conditional Random Fields (CRF) which is supervised is proposed in[1]. This system was developed in 2010 for Arabic language which outperforms the CRF sole work results. HMM based NER system was developed in 2012 for Hindi in[16] which is a dynamic system, language independent and not domain specific. Deepti et al.[8] developed a NER tool in 2013 which can also handle unknown words and performance metrics such as precision, recall and F-measure are calculated.

A multiobjective optimization (MOO) concept was used for selection of feature and optimizing the parameter in[5] by Ekbal and Saha in 2014. It was performed in four classifiers such as maximum entropy, memory based learner, Support Vector Machine (SVM) and Conditional Random Field (CRF) for four different languages like English, Bengali, Telugu and Hindi. The obtained F1 measure in[5] for four languages are 88.68%, 90.48%, 78.71% and 90.44% respectively. Ekbal et al.[7] proposed two methods using algorithm such as ensemble learning and SVM for active annotation where ensemble learning combines SVM and CRF. This system is experimented for English and biomedical text along with Bengali and Hindi language. The ensemble learning approach dominates over the SVM approach.

Pandian et al. in[10] proposed the hybrid approach for NER using E-M algorithm. They consider both POS tags and NER tags as a hidden variable and obtain an average F value for the system as 72.72%. Tamil words provide contextual cues for assigning initial probability which leads to higher precision and recall. Various surveys are done for Named Entity Recognition since NER is a key task for dealing with many NLP tasks such as question answering system, information extraction etc. So Talukdar et al.[17] and Hiremath et al.[15] made a survey on various approaches used for NER for English and Indian languages. They concluded that a hybrid approach combining rule based and statistical technique employing language independent rules should be used to have better performance for NER in Indian languages. NER can be implemented to locate the elements of various categories

based on the domain it is used. One such domain focused application is described in[18]. Vijayakrishna designed the tagset for tourism domain with 106 tags. They considered nested tags and developed a NER system based on CRF. This system consists of three levels of tags and provided the F1 measure of 80.44%. Among all the three levels, system performs well for level-2 tags with F1 measure of 83.77%.

Another NER system based on hybrid approach was developed in[9] by Jayashenbagavalli et al. This system implementation is combination of both rule based and Hidden Markov Model (HMM). Certain rules such as Lexical features and Gazetteer information are imposed. The rule based approach and HMM based approach is used in succession[11] describes about the various challenges encountered while developing a NER system for Indian Languages and more specific to Tamil. This system uses the corpus developed from articles from blogs, newspapers and other web sources with minimal features. It gives the reasonable result for identifying different NE's and unknown entities.

Statistical method uses features for learning the model from training corpus and classifying the test corpus. Feature selection is important in order to classify the entities into particular class in machine learning framework. Asif et al. made an appropriate feature selection for developing Maximum Entropy (ME) based NER system in[6]. Here, they used multiobjective optimization technique to optimize precision and recall. This system determines the suitable feature combinations for Bengali and English yielding precision, recall and F-measure values as 81.88%, 70.76%, 75.91% and 81.27%, 78.38%, 79.80% respectively.

3. Machine Learning Algorithm

3.1 Random Kitchen Sink

Random Kitchen Sink algorithm is a machine learning algorithm for classification of nonlinearly separated data set. The conventional nonlinear kernel methods use large non-linear data set for training the system. It requires large proportion of data points to be stored for classifying new data point. So, space and time requirement is more for classification. Random Kitchen Sink is an alternative for these conventional nonlinear kernel methods. RKS uses only the feature size and does not consider the number of data points for classification.

Random Kitchen Sink algorithm uses GURLS library which is targeted to machine learning, supervised learning to be more specific. It is well suited for large scale machine learning problems especially with multi label problems. Andrea et al.[4] implemented GURLS library in MATLAB which can handle large matrices.

Random Kitchen Sink uses Radial Basis Function (RBK) kernel which is a real Gaussian function. The Fourier Transform of any real Gaussian function which is symmetric is also a real Gaussian function.

$$F(x_1, y_1) = \langle \Phi(x_1), \Phi(y_1) \rangle = e^{\frac{-1}{\sigma}\|x_1 - y_1\|_2^2} \quad (1)$$

$$e^{\frac{-1}{\sigma}\|x_1 - y_1\|_2^2} = e^{\frac{-1}{\sigma}(x_1 - y_1)^T (x_1 - y_1)} \quad (2)$$

$$e^{\frac{-1}{\sigma}(x_1 - y_1)^T (x_1 - y_1)} = e^{\frac{-1}{2}(x_1 - y_1)^T \sum^{-1} (x_1 - y_1)} \quad (3)$$

where

$$\sum = \begin{bmatrix} 2\sigma & 0 & \dots & 0 \\ 0 & 2\sigma & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & 2\sigma \end{bmatrix}$$

The kernel can be expressed in the form as a Gaussian probability density function. Gaussian density function is the product of n Gaussian functions because the covariance matrix is diagonal. The kernel is interpreted as probability density function and associated variables are independent.

Let $x_1 - x_2 = z$ then the kernel function is $f(z) = e^{\frac{-1}{2} z^T \sum^{-1} z}$. Let $F(\Omega)$ represent Fourier Transform of $f(z)$ which is given as

$$F(\Omega) = \frac{1}{2\Pi} \int_{-\infty}^{\infty} f(z) e^{-jz^T \Omega} dz \quad (4)$$

Since $f(z)$ is Gaussian $F(\Omega)$ is again Gaussian. It is a multivariate Gaussian function with variance. $F(\Omega)$ is a Gaussian (multivariate) density function.

$$F^{-1}(\Omega) = \langle \Phi(x_1), \Phi(y_1) \rangle = \int_{-\infty}^{\infty} F(\Omega) e^{jz^T \Omega} dz \quad (5)$$

This can be interpreted as expected value of the quantity $e^{jz^T \Omega}$.

$$E(e^{jz^T \Omega}) = \int_{-\infty}^{\infty} F(\Omega) e^{jz^T \Omega} dz \quad (6)$$

The expected value of any function of random variable is obtained by numerous independent samples from the associated probability density function and an average. Since z is n-tuple, Ω is n-tuple and one particular vector Ω_i can be easily generated.

$$E(e^{jz^T\Omega}) = \frac{1}{k}\sum_{i=1}^{k} e^{jz^T\Omega_i} = \frac{1}{k}\sum_{i=1}^{k} e^{j(x-y)^T\Omega_i} \quad (7)$$

$$\frac{1}{k}\sum_{i=1}^{k} e^{j(x-y)^T\Omega_i} = \frac{1}{k}\sum_{i=1}^{k} e^{jx^T\Omega_i}\overline{e^{jy^T\Omega_i}} \quad (8)$$

3.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm with the given set of examples (inputs) with their labels (outputs). SVM generate a hyperplane which separate two classes. Larger margin of separation, minimizes the error and prediction of new input is likely to be predicted correctly. The two bounding planes are constructed which pass through one or more data points and are parallel. SVM deals with finding a hyperplane which is central between bounding planes by maximizing the margin. Margin is the distance between two bounding planes. The data points that lie exactly on the bounding plane is "support vectors"[19,20]. Let $(x_1, y_1),\ldots, (x_N, y_N)$ be the training samples. x_i is vector of R^N dimension and $y_i \in$ -1, 1 is a label. The two elements of SVM are weight vector w and the bias b which is the distance of hyperplane from origin. The SVM classification rule is

$$\text{sgn}(f(x, w, b)) \quad (9)$$

$$f(x, w, b) =< w.x >+b \quad (10)$$

3.3 Conditional Random Field

Conditional Random Field (CRF) is a statistical method which can be applied on machine learning problems for prediction. CRF is a type of probabilistic model which is used for labeling in Natural Language Processing.

CRF advances than other algorithms by including long range dependencies which avoids biasing problems. By considering the above aggregate predictor model developed as follows.

$$p_\theta(y|x)\alpha \exp\left(\sum_{e\in E,k}\lambda_k f_k(e, y|_e, x) + \sum_{v\in V,k}\mu_k g_k(v, y|_v, x)\right) \quad (11)$$

where x is a word sequence, y is a tag sequence, k is the length of the sequence, $\mu_{y,x}$ and $\lambda_{y,y'}$ is the parameter in the training set[21,22].

4. Experimental Results

This work focuses on identifying the named entities in the given text. Standard NER dataset for Tamil language is not prevalent but the Forum of Information Retrieval and Evaluation, Named Entity Recognition Track (FIRE) dataset is quiet popular among technical community. The size of the dataset[23,24] considered for this work is given in Table 1. The training and test corpus consist of 10003 and 2004 tokens along with their POS tag features respectively. This corpus consists of three labels for named entities 'PERSON', 'LOCATION' and 'ORGANIZATION'. The NER system is developed using different algorithms such as Random Kitchen Sink (RKS), Support Vector Machine (SVM) and Conditional Random Field (CRF). For additional results, the size of the Tamil training corpus is increased to 60k and NER system is developed using Support Vector Machine (SVM) and Conditional Random Field (CRF). The increased dataset size is listed in Table 2.

Table 1. Size of the dataset

Dataset	No. of Words	No. of Sentences	Average Sentence length
Train	10003	773	12.94
Test	2004	179	11.20

Table 2. Size of the Increased dataset

Dataset	No. of Words	No. of Sentences	Average Sentence length
Train	60000	4356	13.78
Test	2004	179	11.20

Random Kitchen Sink algorithm which is implemented in MATLAB does not support text format. So, the dataset is converted into binary matrix which is described in IV-A.

4.1 Binary Matrix Generation

The Random Kitchen Sink algorithms implemented in MATLAB does not support the data set formats like text. So feature extraction process must be done separately. Feature Extraction is transforming the categorical data such as text or images into numerical data which is understandable to machine learning algorithms. This corpus is not used as such for developing a system instead the entire data and theirfeatures are changed into

binary representation of 0's and 1's. The unique features are identified from the whole data set and binary matrix is created.

In this paper, scikit learn module is used for feature extraction by using sklearn.feature_extraction. The class DictVectorizer is used to convert the array of text into Python Dict objects. The categorical features in the corpus are given as "attribute- value" pairs. Using the scikit learn module in Python, both the training and testing data and their features are stored as dictionary and unique characters are identified. We extract the features around each individual token of a corpus which resulted in wide matrix with most of its values are zero and one is being present at the location where the token is present in dictionary. The features that are absent are not stored in the matrix which represents 0.

4.2 Results and Discussions

The total number of entities in Tamil corpus is described in Table 3. The corpus contains 237 'PERSON' entities, 441'LOCATION' and 105 'ORGANIZATION' entities. The name of the place is more when compared to the other entities such as name of person or organization. The total of number of entities in the corpus is 783. The 56.32% of entities are 'LOCATION' and 30.77% of entities are 'PERSON' and remaining 13.41% are 'ORGANIZATION'.

Table 3. Total number of entities in dataset

Entity	Train	Test	Total
Person	204	33	237
Location	379	62	441
Organization	72	33	105
Total	**655**	**128**	**783**

Most of the existing NER systems are implemented using Conditional Random Field (CRF) and Support Vector Machine (SVM). These existing systems (CRF and SVM) are compared with the unique approach carried by applying Random Kitchen Sink (RKS) algorithm which shows almost equal performance with CRF. The dataset is converted into binary matrix for RKS by using scikit learn and this created a matrix with 4690 columns. The size of the matrix created for training and testing corpus is described in Table 4. A vector of length 4690 is generated for each token. The number of tokens is 10,003 which resulted in matrix dimension of 10003 X 4690 binary matrix for training data. The test data resulted in 2004 X 4690 binary matrix.

Table 4. Size of the matrix created from the dataset for RKS

Dataset	Binary Matrix Dimension
Train	10003 x 4690
Test	2004 x 4690

The number of entities identified by the NER system implemented using RKS, SVM and CRF are listed in Table 5. Random Kitchen Sink algorithm recognized 111 entities out of 128 entities in test data. The number of entities identified in the tag 'PERSON', 'LOCATION' and 'ORGANIZATION' are 29, 54 and 28 respectively. The accuracy for each of these entities obtained from RKS are 87.88%, 87.09% and 84.85% as listed in Table 5. The average accuracy is found as 86.61%.

SVM in turn has identified 24 'PERSON', 56 'LOCATION' and 27 'ORGANIZATION' and gives an average accuracy of 81.62%. In case of CRF algorithm, that recognizes 27 'PERSON', 57 'LOCATION' and 29 'ORGANIZATION', an accuracy of 87.21% is seen.

Figure 1 shows the accuracy for various NER systems. Amongst, the accuracy level of 'LOCATION' entity is comparatively better on applying RKS, SVM and CRF. From the analysis, it is evident that RKS shows a good performance in par with CRF.

Table 5. Accuracy of various NER system

Entity	No. of Entity in Test Data	No. of Entities Identified			Accuracy %		
		RKS	SVM	CRF	RKS	SVM	CRF
Person	33	29	24	27	87.88	72.72	81.82
Location	62	54	56	57	87.09	90.32	91.93
Organization	33	28	27	29	84.85	81.82	87.88
Total	**128**	**111**	**107**	**113**	**86.61**	**81.62**	**87.21**

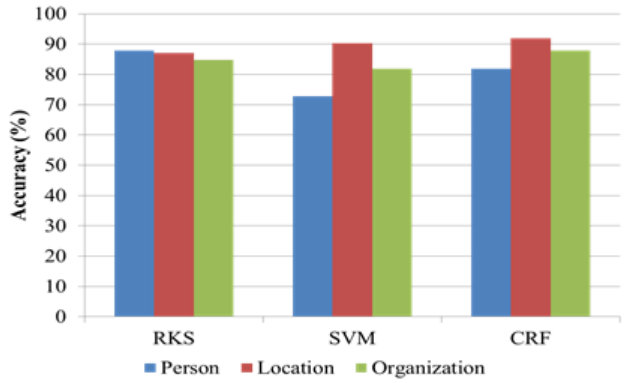

Figure 1. Accuracy of various NER system.

For additional results, Tamil corpus of size 60k was used in order to develop NER system which includes the 54974 tokens for training and 5025 for testing. The number of entities in this dataset is listed in Table 6. Training corpus consists of 2391 'PERSON', 2282 LOCATION' and 453 'ORGANIZATION' entities.

Table 6. Total number of entities in 60k training corpus

Entity	Train	Test	Total
Person	2391	33	2424
Location	2279	62	2359
Organization	453	33	486
Total	**5141**	**128**	**5269**

This corpus is used for implementing NER system based on SVM and CRF. SVM recognizes 28 'PERSON', 53 'LOCATION' and 29 'ORGANIZATION' as described in Table 7 giving an accuracy of 86.06%. For the same entities CRF identifies 28, 57 and 28 respectively. The accuracy given by CRF is 87.20% which is shown in Table 7. Figure 2 gives the accuracy for SVM and CRF for improved corpus.

The accuracy of NER systems developed using SVM and CRF with 10k and 60k corpus differs. The two systems developed using CRF with corpus of 10k and 60k does not show any improvement because of overfitting. Whereas the accuracy for SVM increased about 5.44% for 60k corpus. This intensification is because of increase in size of training corpus with various types of tokens for training than 10k corpus. With the help of this improved training corpus, various types of entities were identified by the system. The time required by the system to obtain the above mentioned accuracy are listed in Table 8.

Table 7. Accuracy for NER system developed using 60k training corpus

Entity	No. of Entity in Test Data	No. of Entity Identified		Accuracy %	
		SVM	CRF	SVM	CRF
Person	33	28	28	84.84	84.84
Location	62	53	57	85.48	91.93
Organization	33	29	28	87.87	84.84
Total	**128**	**110**	**113**	**86.06**	**87.20**

Figure 2. Accuracy of 60k training corpus.

Table 8. Estimated time

Size of Training Corpus	SVM	CRF	RKS
10k	1 min 52 sec	14 sec	14 min 32 sec
60k	3 min 45 sec	4 min 42 sec	-

5. Conclusions and Future Work

In this work, the Tamil dataset consists of only words along with their POS tag feature. The NER system developed using Random Kitchen Sink (RKS) algorithm which recognizes named entities accurately when compare to other machine learning algorithms like SVM and CRF. Random Kitchen Sink algorithm is a unique approach for named entity recognition which can be applied to other languages. In future, RKS can be applied to recognize named entities by including more morphological features. These features will increase the accuracy of the NER system. It can also be applied in order to identify the fine grained entities.

6. References

1. Abdel Rahman S, Mohamed E, Marwa M, Aly F. Integrated machine learning techniques for Arabic named entity recognition. IJCSI. 2010 Jan; 27–36.
2. Rahimi A, Recht B. Random features for large-scale kernel machines. Advances in Neural Information Processing Systems. 2007; 1177–84.
3. Rahimi A, Recht B. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. Advances in Neural Information Processing Systems. 2009; 1313–20.
4. Tacchetti A, Mallapragada PK, Santoro M, Rosasco L. GURLS: A least squares library for supervised learning. The Journal of Machine Learning Research. 2013; 14(1): 3201–5.
5. Ekbal A, Saha S. Multi-objective optimization for classifier ensemble and feature selection: an application to named entity recognition. International Journal on Document Analysis and Recognition (IJDAR). 2012 Jun; 15(2):143–66.
6. Ekbal A, Saha S, Garbe CS. Feature selection using multiobjective optimization for named entity recognition. International Conference on Pattern Recognition; Istanbul. 2010 Aug 23-26. p. 1937–40.
7. Ekbal A, Saha S, Sikdar UK. On active annotation for named entity recognition. International Journal of Machine Learning and Cybernetics. 2014; 1–18.
8. Chopra D, Morwal S, Purohit GN. Hidden markov model based named entity recognition tool. International Journal in Foundations of Computer Science and Technology (IJFCST). 2013 Jul; 3(4):67–73.
9. Jeyashenbagavalli N, Srinivasagan KG, Suganthi S. An automated system for Tamil named entity recognition using hybrid approach. International Conference on Intelligent Computing Applications (ICICA); Coimbatore. 2014 Mar 6-7. p. 435–9.
10. Pandian SL, Krishnan AP, Geetha TV. Hybrid, three-stage named entity recognizer for Tamil. INFOS2008; 2008 Jan.
11. Malarkodi CS, Rao PRK, Lalitha Devi S. Tamil NER- Coping with real time challenges. Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012); 2012. p. 23–38.
12. Padro M, Padro L. A named entity recognition system based on a finite automata acquisition algorithm. Procesamiento del Lenguaje Natural 35; 2005. p. 319–26.
13. Nadeau D, Turney P, Matwin S. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. NRC Publications Archive, Archives des publications du CNRC; 2006.
14. Animesh N, Ravi Kiran Rao B, Singh P, Sanyal S, Sanyal R. Named entity recognition for indian languages. IJCNLP. 2008; 97–104.
15. Hiremath P, Shambhavi BR. Approaches to named entity recognition in Indian languages: A study. International Journal of Engineering and Advanced Technology (IJEAT). 2014 Aug; 3(6).
16. Morwal S, Jahan N, Chopra D. Named entity recognition using Hidden Markov Model (HMM). International Journal on Natural Language Computing (IJNLC). 2012 Dec; 1(4):15–23.
17. Gitimoni T, Borah PP, Baruah A. A survey of named entity recognition in Assamese and other Indian languages. 2014.
18. Vijayakrishna R, Sobha L. Domain focused named entity recognizer for Tamil using conditional random fields. Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages; 2008 Jan. p. 59–66.
19. Gimenez J, Marquez L. SVMTool: A general POS tagger generator based on support vector machines. Proceedings of the 4th International Conference on Language Resources and Evaluation; 2004.
20. Soman KP, Loganathan R, Ajay V. Machine learning with SVM and other kernel methods. PHI Learning Pvt Ltd; 2009 Feb.
21. Lafferty J, McCallum A, Pereira FC. Conditional random fields: Probabilistic models for segmenting and labeling sequence data; 2001.
22. Sutton C, McCallum A. An introduction to conditional random fields for relational learning. Introduction to Statistical Relational Learning. 2006; 93–128.
23. Anand Kumar M, Dhanalakshmi V, Soman KP, Rajendran S. Factored statistical machine translation system for English to Tamil language. Pertanika Journal of Social Science and Humanities. 2014; 1045–61.
24. Dhanalakshmi V, Anand Kumar M, Loganathan R, Rajendran S. Tamil part-of-speech-tagger based on SVM tool. Proceedings of International Conference on Asian Language Processing; Chiang Mai, Thailand. 2009.