




Named-Entity-Recognition (NER) for Tamil Language Using Margin-Infused Relaxed Algorithm (MIRA)

Pranavan Theivendiram, Megala Uthayakumar[✉], Nilusija Nadarasamoorthy, Mokanarangan Thayaparan, Sanath Jayasena, Gihan Dias, and Surangika Ranathunga 

Department of Computer Science Engineering, University of Moratuwa, Moratuwa, Sri Lanka
{pranavan.11, megala.11, nilu.11, mokaranagan.11, sanath, gihan, surangika}@cse.mrt.ac.lk

Abstract. Named-Entity-Recognition (NER) is widely used as a foundation for Natural Language Processing (NLP) applications. There have been few previous attempts on building generic NER systems for Tamil language. These attempts were based on machine-learning approaches such as Hidden Markov Models (HMM), Maximum Entropy Markov Models (MEMM), Support Vector Machine (SVM) and Conditional Random Fields (CRF). Among them, CRF has been proven to be the best with respect to the accuracy of NER in Tamil. This paper presents a novel approach to build a Tamil NER system using the Margin-Infused Relaxed Algorithm (MIRA). We also present a comparison of performance between MIRA and CRF algorithms for Tamil NER. When the gazetteer, POS tags and orthographic features are used with the MIRA algorithm, it attains an F1-measure of 81.38% on the Tamil BBC news data whereas the CRF algorithm shows only an F1-measure of 79.13% for the same set of features. Our NER system outperforms all the previous NER systems for Tamil language.

Keywords: NER · NLP · Tamil · NE · CRF · Margin-Infused relaxed algorithm MIRA

1 Introduction

Named-Entity-Recognition (NER) is a task of identifying named entities in a given text. In other words, NER is used to locate and classify elements in a text into predefined categories such as the names of persons, organizations, locations, date/time, quantities, currency values and percentages. It is a subtask in information extraction. NER is an important precursor in many Natural Language Processing (NLP) tasks such as document summarization, intelligent document access, speech related tasks, machine translation and question answering systems [1].

NER systems built before were based on machine-learning techniques such as HMM (Hidden Markov Model), MEMM (Maximum-Entropy Markov Model), SVM (Support Vector Machine) and CRF (Conditional Random Fields). Among all those, CRF is the most widely used technique. Stanford NER system also uses the CRF algorithm [2]. Most researchers have preferred the CRF algorithm in the past over other multi-class machine learning algorithms such as HMM and MEMM since HMM

suffers from the dependency problem and data sparsity problem and MEMM suffers from label bias problem [3]. CRF gave the best prediction accuracy among the machine learning algorithms.

In addition to the supervised learning techniques, some attempts on building NER systems using semi-supervised learning techniques have been made [4, 5]. Those attempts are based on the active learning approach, which reduces the need of annotated corpus by 80% while maintaining the performance. The success of the system purely depends on informativeness, representativeness and diversity of the selected corpus. Selection of parameters such as batch size and lambda of function for the action learning is quite hard task.

There are 3 existing NER systems for Tamil language. Those systems are implemented using algorithms such Expectation-Maximization [7], SVM [3] and CRF [3, 6]. Previous Tamil NER systems used a limited number of features and these systems are not available for public usage. This has slowed down the Tamil NLP related development. Our research aims to overcome these limitations and to accelerate Tamil NLP related developments by facilitating the identification of 5 different named entities: individual, place, organization, count and time expressions. These tags are selected because person, location, organization, numerical expression and time expression are considered as main Named Entities and increasing the tag set will influence the accuracy of the NER system negatively.

For the named entity classification task, we employ the Margin-Infused Relaxed Algorithm (MIRA). MIRA is a new multi-class algorithm, which has been shown as promising and having potential to be better than most of the other multi-class classification algorithms [8, 9]. This is the first time MIRA is used for Tamil NER. Results show that our MIRA-based NER system gives better performance than all previous Tamil NER approaches. Our system makes use of a basic Part-Of-Speech (POS) tagger and a morphological analyzer (both implemented by us) as an integral part of the system.

This paper consists of five sections. The next section discusses previous work on building NER systems for Tamil. Section 3 describes our approach. Section 4 gives evaluation results of our system, and the final section gives the details about future work and concludes the paper.

2 Related Work

In the past, three different types of techniques have been used for building NER systems. Those are machine-learning techniques [1, 10, 11] grammar based techniques [13] and hybrid based approaches [12]. However, Tamil NER systems have been built only using machine learning techniques [3, 6] and hybrid based approaches [7].

In the context of Tamil NER systems, Vijaykrishna and Shobha [6] built a tourism domain specific NER system for Tamil Language using the CRF algorithm. However they have only used the noun phrases for training and testing. Geetha and Pandian [7] developed a generic NER system for Tamil using the Expectation Maximization (EM) algorithm. Malarkodi et al. [3] created a generic NER system for Tamil language using the CRF algorithm and SVM algorithm separately to compare the performance of the

two algorithms. That research revealed CRF outperforms SVM in the context of Tamil NER. Among those generic NER systems for Tamil language, the best F1-measure obtained so far is 71.68% [3]. None of these Tamil NER systems is available for public usage and some of those systems use Romanization to handle the complexity introduced by some Tamil letters that are made of two or more Unicode code points.

Previous Tamil NER systems used word, POS [3, 6, 7], chunk [3] and patterns (for date and time, in particular) [6] as features.

3 Our Approach

This section describes our approach and the important components of the Tamil NER system. Figure 1 gives the high-level view of the system. Prefix and gazetteer lists are newly used features in our system with respect to the past Tamil NER systems. Our system has three main components: morphological analyzer, part of Speech (POS) tagger and the Named Entity tagger. Morphological analyzer is used to get the stem of the word and a noun/verb tag for the word based on contextual rules.

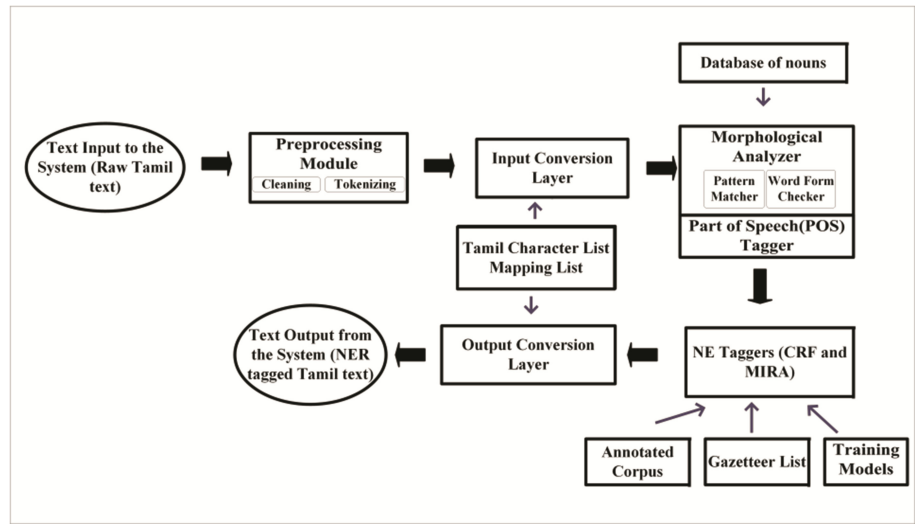


Fig. 1. Overview of the system

3.1 Corpus and Word List Databases

Corpus and the list of words play a vital role in the accuracy and performance of the system. System accuracy increases with the increase of the corpus size.

An annotated corpus of 125,000 words is used for training and testing of the Tamil NER system. This corpus is created from the Tamil BBC newspaper articles.

3.2 Pre-processing

The system uses text files, online resources and websites as an input to the system. It extracts the content and sends them to the preprocessing module. At the preprocessing stage, the content is cleaned and noisy data are removed. Then, the content is tokenized using the Stanford tokenizer [14]. Stanford tokenizer differentiates the sentences by using the “.” as an indicator. However, “.” is used in some other instances other than the end of the sentence. Following are the some usages of “.” other than the end of the sentence:

- Titles - திருமதி (Mrs.), திரு (Mr.)
- Initials - எம்.என்.பெர்னாண்டோ (M. N. Fernando)
- Abbreviation of organization names, political party names, etc. - த.தே.கூ (T. N. A)

In order to overcome above-mentioned misinterpretations in detecting sentences, some rules were incorporated in to the system during the preprocessing phase and the mistakes are corrected.

In Tamil, following are some obvious rules:

1. Sentences do not end with a single letter word
2. Initials and titles are finite and well defined set in Tamil, so we created a list of titles and initials to check those in a given text

Therefore, we incorporated those rules for correcting the problems caused by tokenization.

3.3 Input-Output Conversion Layer

Most of the letters of Tamil Language consist of two Unicode points. Handling one Tamil letter as two Unicode characters gave problems in morphological analysis. Input-Output Layer makes it possible to deal with one Tamil letter as one object. In addition, it gives the opportunity to find the vowel and consonant of compound characters.

For example, க் + ஆ = கா ; if we have the character கா, we can infer consonant க் and vowel ஆ.

As shown in Fig. 1 the raw text is fed to the input conversion layer. After all the NE related processing is done on the objects, the objects are converted back to Unicode format using output conversion layer.

3.4 Morphological Analyzer

Morphological analyzer is a program to analyze the internal structure of the word. It uses different features, language properties and pattern matching. As Tamil is a morphologically rich and agglutinative language (one word affixing with two or more morphemes is known as agglutinative nature), finding the stem of the word is difficult. Most of the words are pinned with morphemes.

A partial morphological analyzer is enough for the NER system. In the context of the Tamil NER system, the morphological analyzer is used for 2 purposes.

1. To identify the stem of the word, in order to reduce the complexity of handling inflected forms of words
2. To derive noun/verb tags for a given word to be used in contextual rules. In addition to that these tags are also used as features for NER algorithms

Our morphological analyzer uses a rule-based approach, which returns only the stem word when the input word is given. It also identifies whether the word is a noun or a verb. For the purpose of rule-based morphological analysis, 100,000 noun stems and verb stems were extracted from the Tamil WordNet using a web crawler. A noun and verb database is created from them. Having all the patterns of a word in the corpus is impossible due to the agglutinative nature of Tamil language. Therefore, the corpus only keeps the stem of the words. Therefore, before checking with the corpus, we find the stem of the word using some rule-based and pattern-matching approaches. Linguistic preprocessing is used to extract the stem from the word.

3.5 Gazetteer and Rule-Based Module

A Gazetteer is created with 100,000 names of people from Sri Lanka. It is a collection of names of Sinhala, Tamil and individuals from other ethnic groups in Sri Lanka. That is, some names are original Tamil names (written originally in Tamil) and others are transliterated from Sinhala (the main language in Sri Lanka). Some of these transliterated names lead to inaccuracies in the Tamil context. For example, the following Sinhala name is conflicting with a Tamil word (which is actually not a name in Tamil). The highlighted portion of the following name caused the contextual problems.

- **இந்து** தரங்கணி சமரரத்ன கொடிகார
(Indhu Tharangani Samararathna Kodikara)
“Indhu” means “Hinduism” in Tamil

Such conflicting names are very rare; therefore, probability of such names occurring in a text of interest is very low. Therefore, such names were removed from the Gazetteer.

The clue words, which are usually useful to find named entities at the first step, are identified and are used to create a rule-based module, which can generate a feature for the Machine Learning model. In the rule-based module, we used Gazetteer of names and clue words for Person, Organizations, Places, and Count and compared them with the text to be tagged. If the clue words are found in the given text, we indicate that as a feature to the machine learning based tagger.

Organizations have about 100 clue words, which are usually used at the end of the organization names.

Eg: கூட்டுத்தாபனம் (Corporation), திணைக்களம் (Department)

For persons, we have used title and salutation words. The following are some example of title words.

- திரு (Mr), திருமதி (Mrs), செல்வி (Miss)

There will be a high chance that a name of a person following a title word. For places, we have added the main cities and names of all the countries of the world to the clue words list of places. For count (numbers), literals used to represent numbers and decimals are added to the clue words list.

3.6 Part-of-Speech (PoS) Tagger

Part Of Speech tagging is marking up a word in a text with a corresponding predefined part of speech tag. PoS is a useful feature in NER because most of the named entities, especially names are proper nouns. Hence, there is a high chance that a noun can be a Named Entity.

Stanford PoS tagger [15] is used to build a Tamil PoS tagger. The standard PoS definition has 32 tags [16]. Classification into higher number of classes ended up in higher error rates. Hence, tags set is defined in manner where 21 tags are used. 80,000 words are tagged manually and used as training data for the tagger. These tags are NN, NNC, RB, VM, SYM, PRP, JJ, NNP, PSP, QC, VAUX, DEM, UT, QF, NEG, QO, CC, WQ, INTF, NNPC and RBP¹.

3.7 Named-Entity Tagger

This part is implemented using the MIRA classifier. We also implemented it using CRF as well, in order to compare with the performance of MIRA. Both the classifiers are trained using CRF++ tool [17]. Following are the tags used for tagging the named entities in the given text.

- INDIVIDUAL
- PLACE
- ORGANIZATION
- TIME
- COUNT

The following are the features considered for building the classifiers.

- Contextual length (window size)
- Part-of-Speech (POS) tag
- Noun and verb phrases derived from the morphological analyzer
- Gazetteers
- Surefire rules

Example: We have used the following as surefire rules in order to facilitate the identification of the NEs in a text.

¹ NN - Noun, NNC - Compound Noun, RB - Adverb, VM - Verb Main, SYM - Symbol, PRP - Personal Pronoun, JJ - Adjective, NNP - Pronoun, PSP - Prepositions, QC - Quantity Count, VAUX - Verb Auxiliary, DEM - Determiners, QF - Quantifiers, NEG - Negatives, QO - Quantity Order, WQ - Word Question, INTF - Intensifier, NNPC - Compound Pro Noun.

Surefire rules:

- For individuals: there is a high chance that a name of an individual follows after titles such as Mr., Mrs., Prof., Dr.
- For organizations: there is a set of starting and ending words such as station, department, organization, university and so on.
- Prefixes and suffixes
 - E.g. Prefix for “University”: U, Un, Uni,..
 - Suffix for “University”: y, ty, ity,....

Some NEs start with certain prefix. For example, சிவகாமி (Sivagamy), சிவநாதன் (Sivanathan), சிவநேசன் (Sivanesan) are some names which starts with the same prefix “சிவ (Siva)”. Similar logic applies for suffix as well.

- Orthographic features

Orthographic features are like a pattern, which match a tokenized word. In the following examples, “X” is used as a placeholder to represent Tamil characters whereas digits are denoted with the letter “N”. Other special characters are used as it is.

E.g.: மாலா	→ XX
போனாள்	→ XXX
45	→ NN

In particular, orthographic feature is very helpful in identifying date and time. For instance, NN-NN-NNNN pattern shows it is a date.

- Length of word

Gazetteer, surefire rules, prefix, suffix, orthographic features and length of words are novel features used by us (no other Tamil NER systems used them). These features are selected based on previous successful researches on other languages [1, 8]. Part of our research involves identification of best feature combination for Tamil NER system. However, we were not able to use some of the features used in other languages due to language and resource constraints. Consider following two examples.

1. Most of the English NER systems use capitalized first letter of a word as a feature for their NE systems as English has a capitalization concept for proper nouns. However, it is not applicable for Tamil language, which does not have the capitalization concept.
2. Stanford NER [2] uses distributed similarity feature, which is based on similarity between words. In order to use this feature, an annotated clustered corpus is required. However, that type of resource is not available for Tamil language.

3.8 MIRA Based Tagger

MIRA has been proven to be better than CRF in the context of some other languages [9], and for some other languages CRF and MIR performs equally [18]. Bengali Named Entity Recognition system had a better performance when using MIRA algorithm rather

than CRF when tested using South and South East Asian Languages (NERSSEAL) shared task data. MIRA based English Named Entity Recognition system had similar performance as CRF algorithm when tested with CoNLL-2003 data set. Tamil belongs to Dravidian language family while Bengali belongs to Indo-Aryan family and English belongs to Indo-European family, but Dravidian languages share strong areal feature with Indo-Aryan languages [19] and they do not show any significant connection with Indo-European languages. Therefore, there is a high chance that Tamil NER may show similar performance as Bengali NER. Therefore, we decided to test the performance of MIRA algorithm in the context of Tamil NER.

MIRA is an online algorithm, which is based on error minimization. It makes use of a matrix to build a model. In each iteration, different matrices are considered by making a small change to the parameters of the earlier matrix and the matrix that makes the lowest error is selected as the final matrix. Likewise, iterations are continued throughout the training data and final matrix is discovered. Pseudo code of MIRA algorithm [8] is given below,

Initialize: Set $M \neq 0, M \in R^{k \times n}$

Loop: For $t = 1, 2, \dots, T$

– Get a new instance \bar{x}^t

– Predict $\hat{y}^t = \operatorname{argmax}_r \left\{ \overline{M}_r \cdot \bar{x}^t \right\}$

– Get a new label y^t

– Find $\bar{\tau}^t$ that solves the following optimization problem:

$$\min_{\tau} = \frac{1}{2} \sum ||\overline{M}_r + \tau_r \bar{x}^t||_2^2$$

Subject to : (1) $\tau_r \leq \delta_{r,y^t}$ for $r = 1, \dots, k$

$$(2) \sum_{r=1}^k \tau_r = 0$$

– Update: $\overline{M}_r \leftarrow \overline{M}_r + \tau_r^t \bar{x}^t$ for $r = 1, 2, \dots, k$

Output = $H(\bar{x}) = \operatorname{argmax}_r \left\{ \overline{M}_r \cdot \bar{x} \right\}$

3.9 Conditional Random Field (CRF) Based Tagger

Another classifier used by this system is based on the CRF algorithm. We have selected CRF since it has shown good performance over other techniques in Tamil NER as stated before.

Lafferty et al. [20] defines the CRF algorithm as follows:

Let $G = (V, E)$ be a graph such that, $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G . Then (X, Y) is a conditional random field in case, when conditioned on X , the random variables Y_v obey the Markov property with respect to the graph:

$p(Y_v | X, Y_w, w \neq v) = p(X, Y_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G .

The joint distribution over the label sequence Y given X has the form,

$$p_{\theta} \propto \exp \sum_{e \in E,k} \lambda_k f_k(e, y|_e, X) + \sum_{e \in V,k} \mu_k g_k(v, y|_v, X)$$

Where x is a data sequence, y a label sequence, and $y|_S$ is the set of components of y associated with the vertices in sub graph S . CRF predicts the label for the given data based on the probability of a label given the features.

4 Evaluation

The POS Tagger was trained using the 80,000 words from the Tamil BBC News. Precision measures for POS tagger is given in Table 1.

Table 1. Result of the POS tagger

Number of words	Correct predictions	Precision %
747	707	94.65
1,169	1,104	94.44
1,567	1,473	94.00

125,000 words collected from Tamil BBC News are used as the data set for Tamil NER system. Evaluations were performed using 10-fold cross validation. For each instance of testing, 112,500 words are used for training the model and the testing is done using the remaining 12,500 words. Table 2 gives the comparison of MIRA algorithm and CRF algorithm with respect to various features.

Table 2. Comparison of MIRA and CRF algorithms

Features used	F1 - measure	
	MIRA	CRF
With only word features	61.0%	46.6%
Word features + POS	72.20%	68.34%
Word features + noun and verb tags derived from morph	72.25%	57.37%
Word features + POS + noun and verb tags derived from morph	70.93%	67.37%
Word features + POS + noun and verb tags derived from morph + gazetteer + surefire rules	78.45%	77.53%
Word features + POS + noun and verb tags derived from morph + gazetteer + surefire rules + suffix	76.73%	76.15%
Word features + POS + noun and verb tags derived from morph + gazetteer + surefire rules + prefix	80.73%	79.13%
Word features + POS + noun and verb tags derived from morph + gazetteer + surefire rules + orthographic features	81.38%	76.76%

It is clear that the MIRA algorithm outperforms the CRF algorithm in most of the instances, based on the overall F1-measure of the system. With the above testing, we

found that the optimal features for MIRA model are window size 3, PoS, noun and verb tags derived from morph, gazetteers, surefire rules, prefix of length 4 and orthographic features. Optimal features for CRF model are window size 3, POS tag, noun and verb tags derived from morph, gazetteer, surefire rules and prefix length 2. Our tests revealed that the suffix feature is not suitable to be used in conjunction with the gazetteer feature as it negatively affects the F1-measure. Table 3 gives the evaluations results of individual entities for CRF and MIRA models with the best feature combinations of respective algorithms.

Table 3. Comparison of MIRA and CRF algorithm for different entities

Named entity	Precision (in %)		Recall (in %)		F1-measure (in %)	
	MIRA	CRF	MIRA	CRF	MIRA	CRF
INDIVIDUAL	85.12	86.00	84.68	77.36	84.90	81.45
ORGANIZATION	85.92	93.33	67.78	62.22	75.78	74.67
COUNT	90.45	82.84	78.92	82..84	84.29	82.84
PLACE	95.30	93.94	69.25	67.39	80.22	78.48
TIME	95.08	100	71.60	64.20	81.69	78.20
OVERALL	90.37	91.22	74.45	70.80	81.38	79.13

With the above table, it is very clear that MIRA outperforms CRF in most of instances and the test results reveal that when making predictions, MIRA focuses more on recall and the CRF focuses more on precision.

Table 4 gives the comparison of our approach with the previous attempts. Our system shows an increase of 9% in F1-measure with respect to previous generic NER systems and our system outperforms the domain focused NER system with a small margin.

Table 4. Comparison of our approach against previous Tamil NER systems

Approaches	Precision	Recall	F1-measure
Shobha and Vijakrishna's tourism domain NER (2008)	88.52%	73.71%	80.44%
Geetha and Pandian's generic NER (2008)	83.01%	64.70%	72.70%
Malarkodi et al. generic NER (2012)	71.28%	70.51%	70.68%
Our generic NER (2016)	90.37%	74.45%	81.38%

5 Conclusion

This paper presented a Named Entity Recognition (NER) system for Tamil language using MIRA. By trying out different features, we have found the optimum combination of features to get a F1- measure of 81.38%. Optimum set of features are word interval, PoS tag, noun and verb tags derived from morph, gazetteer list, sure-fire rules, prefixes and orthographic features. When compared with CRF, the algorithm traditionally used for Tamil NER, MIRA classifier out-performs it in most of the instances for Tamil NER.

As future work, we expect to support more Named Entity types such as currency and percentage. In addition to that, we expect to expand the corpus and to identify further features in order to increase the accuracy of the system.

Acknowledgement. We would like to thank AU-KBC research centre of Chennai, Forum for Information Retrieval Evaluation (FIRE) and Department of Registrations of Persons Sri Lanka for providing us necessary language resources and tools to carry out this research.

References

1. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. *Linguisticae Investig.* **30**(1), 3–26 (2007)
2. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 363–370 (2005)
3. Malarkodi, C.S., Pattabhi, R.K., Sobha, L.D.: Tamil NER—coping with real time challenges. In: *24th International Conference on Computational Linguistics*, pp. 23–38 (2012)
4. Laws, F., Schätze, H.: Stopping criteria for active learning of named entity recognition. In: *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pp. 465–472 (2008)
5. Shen, D., Zhang, J., Su, J., Zhou, G., Tan, C.-L.: Multi-criteria-based active learning for named entity recognition. In: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, p. 589 (2004)
6. Vijayakrishna, R., Sobha, L.: Domain focused named entity recognizer for tamil using conditional random fields. In: *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages*, pp. 59–66 (2008)
7. Pandian, S., Pavithra, K.A., Geetha, T.: Hybrid three-stage named entity recognizer for tamil. In: *The Sixth Annual Conference on Informatics and Systems (INFOS)*, pp. 45–52 (2008)
8. Crammer, K., Singer, Y.: Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.* **3**, 951–991 (2003)
9. Banerjee, S., Naskar, S.K., Bandyopadhyay, S.: Bengali named entity recognition using margin infused relaxed algorithm. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) *TSD 2014. LNCS (LNAI)*, vol. 8655, pp. 125–132. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10816-2_16
10. Cunningham, H., Tablan, V., Roberts, A., Bontcheva, K.: Getting more out of biomedical documents with GATE’s full lifecycle open source text analytics. *PLoS Comput. Biol.* **9**(2), e1002854 (2013)
11. Bird, S.: NLTK: the natural language toolkit. In: *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pp. 69–72 (2006)
12. Ekbal, A., Haque, R., Das, A., Poka, V., Bandyopadhyay, S.: Language independent named entity recognition in indian languages. In: *IJCNLP*, pp. 33–40 (2008)
13. Mikheev, A., Moens, M., Grover, C.: Named entity recognition without gazetteers. In: *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, pp. 1–8 (1999)
14. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The stanford CoreNLP natural language processing toolkit. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55–60 (2014)

15. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pp. 173–180 (2003)
16. Dhanalakshmi, V., Shivapratap, G., Soman Kp, R.S.: Tamil POS tagging using linear programming. *Int. J. Recent Trends Eng.* **1**(2), 166–169 (2009)
17. Kudo, T.: CRF++: Yet another CRF toolkit, CRF++: Yet Another CRF toolkit (2005). <https://taku910.github.io/crfpp/>. Accessed 24 Jan 2016
18. Crammer, K., McDonald, R., Pereira, F.: Scalable large-margin online learning for structured classification. In: NIPS Workshop on Learning With Structured Outputs (2005)
19. Krishnamurti, B.: The Dravidian Languages. Cambridge University Press, Cambridge (2003)
20. Lafferty, J., McCallum, A., Pereira, F.C.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML 2001 Proceedings of the Eighteenth International Conference on Machine Learning, pp. 282–289 (2001)