CUSAT_TEAM@IECSIL-FIRE-2018: A Named Entity Recognition System for Indian Languages

Ajees A P[1] and Sumam Mary Idicula[2]

[1] Research Scholar, CUSAT, Cochin 682022, INDIA
ajeesap87@gmail.com
[2] Professor, CUSAT, Cochin 682022, INDIA
sumam@cusat.ac.in

Abstract. Named Entity Recognition is the process of classifying the elementary units in a text document into meaningful categories such as person, location, organization, etc. It is a significant preprocessing step in the semantic analysis of natural language text. There is an enormous growth of Indian language content on various media types such as websites, blogs, email, chats, etc. over the past decade. Automatic processing of this huge unstructured data is a challenging task especially when the companies are interested to ascertain public view on their products and processes. NER is one of the subtasks of Information Extraction. Extracting structured information from the natural language text is the ultimate goal of IE systems. Different methods are proposed and experimented for NER. In this paper, we propose a Named Entity Recognition system for Indian languages using Conditional Random Fields. Training and testing are conducted using the shared corpus provided by 'ARNEKT-IECSIL 2018' competition organizers. The evaluation results show that the proposed system is able to outperform most of the reported methods in the competition.

Keywords: Named Entity Recognition · Conditional Random Fields · Natural Language Processing · Supervised learning.

1 Introduction

The Internet is the fastest growing resource on the world. Lots of information are added to the web every second. But this information is stored in an unstructured manner. Retrieving the relevant information from this unstructured text is a challenging task that invites the focus of Language researchers. Information extraction, a branch of Artificial Intelligence deals with this challenge [11]. IE transforms the unstructured text into a structured form that can be easily handled by machines. Named Entity Recognition is one of the subdomains of IE. It is the process of identifying a word or phrase that refers to a particular entity within a text. The term entity is coined in the Sixth Message Understanding Conference (MUC) [6]. Most of the benchmarks in NER are also reported from MUC conferences. Recognizing the semantically meaningful classes of words from an unstructured text is the goal of NER systems. Even though different solutions are reported for the problem, it is still an open area of research.

Categorizing the articles according to the content helps in smooth content discovery. NER systems can automatically scan the articles and identify the important entities mentioned in them. Knowing the relevant tags for articles can help in automatic categorization of the articles and hence easy content discovery [10]. NER systems can also be used to empower the searching algorithms. Most of the online publications have millions of articles in their database. Searching the complete list of articles for all the queries will take enormous time. Tagging all the articles with relevant entity tags and storing that tags separately can speed up the search operation to a considerable extent. Content Recommendation is another application where NER systems can be utilized. Extracting the entities from the viewed articles and recommending other articles with similar entities can improve the recommendation systems. NER systems also help in identifying the position of the text that should be transliterated rather than meaning translated.

The structure of this article is as follows. Section 2 briefly reviews the related works. Section 3 explains the task description and details about the dataset. Section 4 discusses the methodology and section 5 illustrates the experiments and results. Finally, section 6 concludes the article along with some routes for the future works.

2 Related Works

The term named entity refers to a word or phrase that clearly distinguish one item from the other set of items. MUC-6, where the term named entity is introduced categorize entities into 3 classes namely- ENAMEX, TIMEX, and NUMEX. ENAMEX comprises entities like person, location, organization, etc. Date and time are included in TIMEX. NUMEX covers entities like money, quantity, and percentage. Mainly three types of approaches are reported in NER. They are supervised, semi-supervised and unsupervised approaches. Supervised methods try to build a model by looking at annotated training examples. Here a set of features are used to represent a word in the training data. These features form the input to the learning algorithm. The tags of words act as supervisors to fine tune the model parameters. Hidden Markov Model, Maximum Entropy Markov Model, SVM model, etc. are some of the models employed in such studies [13].

The major motivation towards the semi-supervised learning algorithms is the lack of enough labeled data. Semi-supervised learning algorithms make use of both labeled and unlabeled data to create their own hypothesis. They start with a small amount of labeled data and continue with a large amount of unlabelled corpus to build the classifier. Here more annotations are generated iteratively until a threshold is reached. NER using Adaboost is an example of semi-supervised NER system [7].

In order to overcome the requirements of supervised learning algorithms, unsupervised learning algorithms are introduced. Supervised learning methods demand a robust set of features and a large amount of annotated corpora. 'KNOWITALL', proposed by Etzioni et al. is a pillar example of unsupervised Named Entity Recognition system [8]. It is a domain-independent system that makes use of domain-independent extraction patterns to generate candidate facts.

When it comes to Indian languages, the major challenge in NER are as follows. The capitalization feature is absent in almost all the Indian languages [9]. Whereas the other languages like English make use of capitalization feature in the identification of named entities. The morphological richness of the word forms is another problem in Indian languages, which makes it difficult to identify root words from its inflected forms [14]. Ambiguities at word level is also a challenge in the identification of named entities. The same word can act as an entity or a common noun in different contexts. Most of the Indian languages are free word order languages, which affect n-gram based approaches of NER [3]. Spelling variations in names create another hindrance to the problem of NER in Indian languages. The same word is spelled in different ways by different peoples. All these issues get accumulated and make the problem of NER in Indian languages a challenging one.

CRFs are very promising in entity recognition task. Even the best performing Stanford entity tagger is based on CRFs. They are not novel to the field of NER in Indian languages. Many works are reported in different South Asian languages including Tamil, Hindi, Telugu, Malayalam etc. Most of them are based on the personal and limited dataset which is the major bottleneck in their works. Sharma et al. [15] , Srikanth et al. [16], Prasad et al. [12] and Vijayakrishna et al. [17] are some of the works reported in Indian languages using CRF.

3 Task Description and Dataset Details

The shared task is divided into two subparts say task-A and task-B [5]. Task-A deals with the identification of named entities from the raw text and task-B deals with extracting relation amongst the entities in a sentence. Both these tasks come under the domain of Information Extraction (IE), which is an area under constant research. The growth of research in this area leads to the advancement of applications like information search, question answering, document summarization, etc. Five Indian languages are considered for this shared task. They are Tamil, Hindi, Kannada, Telugu, and Malayalam. It is well known that IE works significantly well with languages like English from applications like Google search, frameworks like Stanford CoreNLP, OpenNLP and many more. The same does not hold good for Indian Languages due to its morphologically rich nature and agglutinative structure. Hence the goal of this task is to improve the Information Extraction systems for Indian languages [2].

The shared dataset contains data from five different Indian languages [4]. The training data for task-A is a set of files in plain text format. Each file consists of words and their labels in a line by line basis. Each language has more than five lakhs samples of training data. Statistics of the training data for task-A is shown in table 1. The testing data contains two files say test1 and test2. Test1 is for pre-evaluation and test2 is for final evaluation.

Table 1. Training data statistics

Language	# sentences	# words	# unique words
Hindi	76537	1548570	88198
Tamil	134030	1626260	186267
Malayalam	65188	903521	145240
Telugu	63223	840908	108224
Kannada	20536	318356	73836

4 Proposed method

The proposed system is a CRF-based sequence labeling model with words as the input sequence and entity tags as output sequence. CRFs are probabilistic graphical models used for labeling sequential data. They can be used to predict any sequence in which multiple variables depend on each other. A key advantage of CRFs over other sequence labeling models is their great flexibility to include a range of arbitrary and dependent features of the input. Since Indian languages are morphologically rich, a wide variety of such morphological features can be used to enrich the input word representation. Figure 1 shows the graphical illustration of CRF. Here each vertex represents a random variable and each edge represents the association between the random variables. CRFs are free from label bias problem, a weakness exhibited by Maximum Entropy Markov Models. They are capable of producing multiple variables that are mutually dependent. Let $W = w_1, w_2, w_3, ...w_n$ be the input sequence and $Y = y_1, y_2, y_3, ...y_n$ be the corresponding label sequence. CRFs try to maximize the conditional probability distribution$P(Y/W)$ given the input sequence. The best entity tag sequence corresponding to a word sequence is calculated as shown in equation 1.

$$\hat{\bar{y}} = \arg\max_{\bar{y}} P(\bar{y} \mid \bar{W}; \bar{w}) \tag{1}$$

Here $\bar{W}$ is the observable word sequence and $\bar{y}$ is the corresponding hidden entity tag sequence. The probability of a tag sequence $\bar{y}$, for a given word sequence $\bar{W}$, is calculated as in equation 2. Where $\bar{w}$ indicates the weight vector and 'F' indicates the global feature vector.

$$P(\bar{y} \mid \bar{W}; \bar{w}) = \frac{\exp(\bar{w} \cdot F(\bar{W}, \bar{y}))}{\sum\limits_{\bar{y}' \in Y} \exp(\bar{w} \cdot F(\bar{W}, \bar{y}'))} \tag{2}$$

The conditional probability of Y_i on W is defined by a set of feature functions. Each feature function is assigned by a particular weight as shown in equation

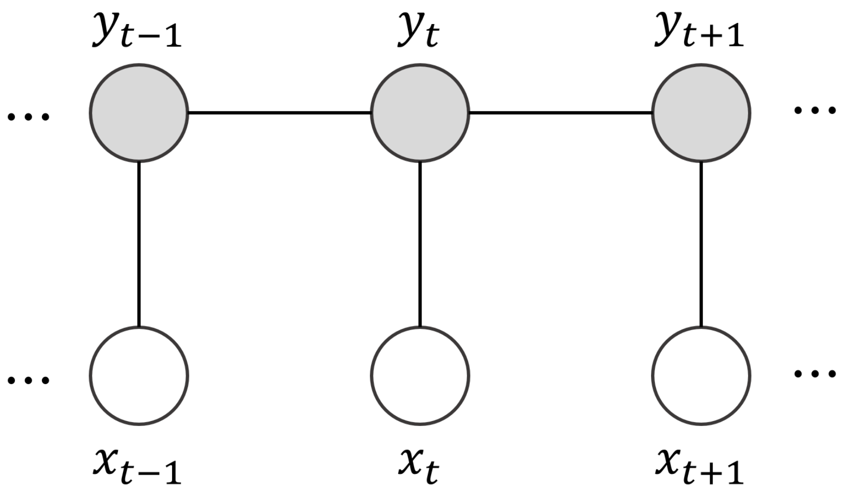

Fig. 1. CRF: A graphical illustration

3. The feature functions can inspect the entire input sequence W at any point during the inference. Each feature function can analyze the entire observation sequence $\bar{W}$, the current y_i and previous y_{i-1} positions in the tag sequence and current position 'i' in the observation sequence. A feature function is computed by summing f_k over all n different state transitions $\bar{y}$.

$$F(\bar{W}, \bar{y}) = \sum_i f(y_{i-1}, y_i, \bar{W}, i) \tag{3}$$

Finally, the best tag(named entity) sequence is decoded using the Viterbi algorithm.

5 Experiments and Results

The architecture of the proposed system is shown in figure 2. The first stage of the architecture is the preprocessing stage, where the tagged text is converted into sequences of words and sequences of tags. In the second stage, each word from each sentence is sent to a feature preparation module, where features for each word is prepared. Hence the sequences of words are converted into sequences of features. The different feature we have considered for CRF training is the word, preceding words, following words, suffixes of different length, number information, length information, etc. The labels of words are also converted into sequences of tags to facilitate CRF training. The third phase of the architecture is the training phase, where the model parameters are learned. We have used Pycrfsuite, a python based implementation of CRF for training [1]. Training is performed on the tagged data for 50 epochs and the model is saved. The final phase of the architecture is the testing phase, where the performance of the

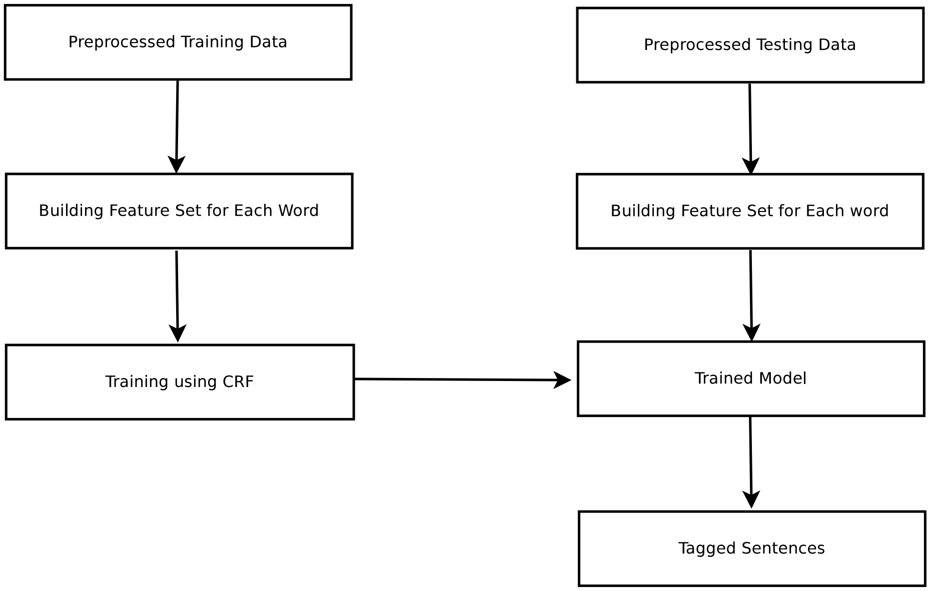
Preprocessed Training Data
Building Feature Set for Each Word
Training using CRF
Preprocessed Testing Data
Building Feature Set for Each word
Trained Model
Tagged Sentences

Fig. 2. Architecture of the proposed system

model is assessed. Twenty percent of the total data is used for testing. The words in the test data are also preprocessed as in the training data.

The proposed system is tested with two test datasets(pre-evaluation and final evaluation). Our system predicts the label sequence for each input sentence. Table 2 demonstrates the results of our system on both the datasets. It is clear from the results that our system performance is promising as compared with the performance of other methods reported in this competition.

Table 2. Results

Test data	Hindi	Kannada	Malayalam	Tamil	Telugu	Average
Test 1 (Accuracy %)	97.67	97.03	97.44	97.36	97.72	97.44
Test 2 (Accuracy %)	97.65	97.09	96.86	96.85	97.69	97.23

6 Conclusion

In this paper, we have discussed a CRF based Named Entity Recognition system for Indian languages. The exclusive feature of this approach is its performance as compared to other sequence labeling techniques. The main reason we preferred CRFs rather than traditional statistical methods is their ability to model the sequence to sequence learning problems. Since CRFs are statistical models, the performance of the system can be improved by increasing the training data size. The performance of the system can also be improved by incorporating word embedding based cluster features into the CRF training. Due to the lack of enough computational resources, we could not execute that operation. Apart from NER, Conditional Random Fields can also be applied to various NLP applications like POS tagging, semantic role labeling, word phrase chunking, etc.

References

1. A python binding for crfsuite. https://github.com/scrapinghub/python-crfsuite, accessed: 2017-09-30
2. Arnekt Solutions: Information extractor for conversational systems in indian languages (2018), https://iecsil.arnekt.com, [Online; accessed 14-July-2018]
3. Athavale, V., Bharadwaj, S., Pamecha, M., Prabhu, A., Shrivastava, M.: Towards deep learning in hindi ner: An approach to tackle the labelled data scarcity. arXiv preprint arXiv:1610.09756 (2016)
4. Barathi Ganesh, H.B., Soman, K.P., Reshma, U., Mandar, K., Prachi, M., Gouri, K., Anitha, K., Anand Kumar, M.: Information extraction for conversational systems in indian languages - arnekt iecsil. In: Forum for Information Retrieval Evaluation (2018)
5. Barathi Ganesh, H.B., Soman, K.P., Reshma, U., Mandar, K., Prachi, M., Gouri, K., Anitha, K., Anand Kumar, M.: Overview of arnekt iecsil at fire-2018 track on information extraction for conversational systems in indian languages. In: FIRE (Working Notes) (2018)
6. Bindu, M., Idicula, S.M.: Named entity identifier for malayalam using linguistic principles employing statistical methods. International Journal of Computer Science Issues(IJCSI) **8**(5) (2011)
7. Carreras, X., Màrquez, L., Padró, L.: A simple named entity extractor using adaboost. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. pp. 152–155. Association for Computational Linguistics (2003)
8. Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. Artificial intelligence **165**(1), 91–134 (2005)
9. Goyal, A., Kumar, M., Gupta, V.: Named entity recognition: Applications, approaches and challenges
10. ParallelDots: Named entity recognition: Applications and use cases (2016 (accessed July 7, 2018))
11. Patil, N., Patil, A.S., Pawar, B.: Survey of named entity recognition systems with respect to indian and foreign languages. International Journal of Computer Applications **134**(16) (2016)
12. Prasad, G., Fousiya, K., Kumar, M.A., Soman, K.: Named entity recognition for malayalam language: A crf based approach. In: Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on. pp. 16–19. IEEE (2015)
13. Sasidhar, B., Yohan, P., Babu, A.V., Govardhan, A.: A survey on named entity recognition in indian languages with particular reference to telugu. International Journal of Computer Science Issues (IJCSI) **8**(2), 438 (2011)

14. Shah, H., Bhandari, P., Mistry, K., Thakor, S., Patel, M., Ahir, K.: Study of named entity recognition for indian languages. Int. J. Inf **6**(1), 11–25 (2016)
15. Sharma, R., Goyal, V.: Name entity recognition systems for hindi using crf approach. In: Information Systems for Indian Languages, pp. 31–35. Springer (2011)
16. Srikanth, P., Murthy, K.N.: Named entity recognition for telugu. In: Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages (2008)
17. Vijayakrishna, R., Sobha, L.: Domain focused named entity recognizer for tamil using conditional random fields. In: Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages (2008)