

Named Entity Recognition in Sanskrit: A Survey

¹Joseph Mickole James, ²Sangeetha Jamal

¹Dept. of Computer Science, Rajagiri School of Engineering and Technology

Email: josephmj197@gmail.com

² Asst. Professor, Dept. of Computer Science, Rajagiri School of Engineering and Technology

Email:sangeethajamal2010@gmail.com

Abstract

This paper gives an overview of the various methods used for Named entity recognition in Sanskrit language. Named entity recognition (NER) is the process of recognizing entities (e.g. Person, Place, Organization etc) associated with a particular word of a document. NER has a significant part in Natural Language Processing applications like Information Retrieval, Machine Learning, Survey systems, automatic summarizing etc. It is also used to organize every word in a document into predefined categories. NER systems use language grammar based techniques as well as arithmetical models. Since entity names are the major part of the document, NER is a key step towards more intellectual data mining and its management. Six different approaches are studied and compared and a conclusion is drawn.

Keywords: Natural Language Processing, Sanskrit, Named Entity Recognition, NER, Indian Languages, Fuzzy Support Vector Machine

1. Introduction

Named entities in a text are the atomic elements that represent the name of something, and the name can be a person name, name of an organization, name of a place or location etc. In the field of information extraction, the identification and classification of named entities are quite an important task. The identification and classification of the named entities in a text to some predefined classes is known as named entity recognition. The commonly used predefined classes are the person name, place, name of organization, date, numerical data, measurements, and so on.

1.1 Motivation

The named entity recognition task involves identification of proper nouns in texts and their classification into a set of predefined categories of interest. Although over the years there has been considerable work done for NER in English and other European languages, the interest in Indian languages has been quite low. Sanskrit has rich morphology compared to other languages. Over the past decade Indian language content on various media types such as websites, blogs, and emails has increased significantly. A method to process this huge data automatically is needed since companies are interested to ascertain a public view on their products and processes. This requires natural language processing software systems which identify entities, identification of associations or relation between entities.

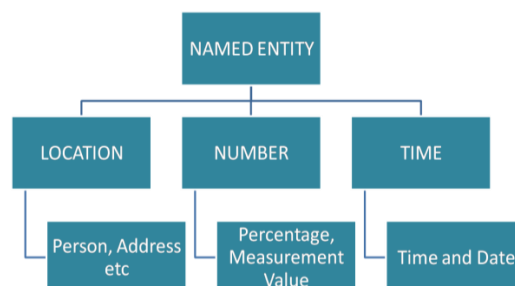


Figure 1: Named Entity Types [13]

NER plays a vital role in reference resolution, disambiguation, representation of meaning etc, in other natural language processing applications. Semantic parsers and part of speech taggers can be extended with this type of tagging to provide better results. An understanding of named entities involved in a text or document at any level will provide efficient analytical frameworks and cross-referencing. Since Sanskrit is the base language of numerous Indian and European languages this system can be enhanced to work for other languages too.

2.Related Works

Various works on NER in Sanskrit is available in recent years. This is the result of Sanskrit being digitized in past few years. This section will summarize such works.

Named Entity Recognition and Classification for Gujarati Language [10]

The main focus is on Gujarati Language. Gujarati Language is rich in terms of entities or words. Each character is different from the other in terms of shape, size and other dimensions. The main focus of the paper is to recognize Gujarati named entities (NE) from a group of words and classify each word based on the different classes like Name, Location, and Date. To achieve this, NER is implemented using Hidden Markov Model (HMM).

Semi-Supervised Learning Approach for Indonesian Named Entity Recognition Using Co-Training Algorithm [9]

In Indonesian NER, the machine learning approach supervised by a learning model tends to have better results. Even though the supervised learning model gives better results, it has limited labeled data for the training process. The design system of this paper has two parts. The first part is the co-training process. The purpose of this part is to collect new labeled data from unlabelled data. The second part is testing the new NER system. The new NER system utilizes new labeled data from the co-training process. In both the processes, a support vector machine is used in the classification process. Two types of data are used, labeled data and unlabelled data which is used as the supervised data and semi supervised learning data respectively.

Named-Entity Recognition in Bengali[8]

Conditional Random Fields are used for the NER task in Bengali. CRF++ is used, which is an open-source package. Three different CRF-based classifiers are used for predicting the three levels of a named-entity. Each level takes the output from preceding level as its input and it is added as a feature for the next level of classification. Language independent features are used in different combinations and finally those features for which the highest accuracy is obtained for a development set is chosen. The development set is created by randomly selecting a set of sentences from the training data.

Arabic Named Entity Recognition Using Artificial Neural Network [3]

This system is designed based on neural network approach. The main task of a neural network is to automatically learn to recognize patterns and make smart decisions using the data available, so that it can be applied to classify new texts within large databases. Neural network approach has proved to be very successful in many areas of artificial intelligence. The proposed system consists of three stages: the Pre-processing stage that cleans the data, the second is Conversion which converts Arabic letters to Roman alphabets and the third stage uses neural networks to classify the input data.

Named Entity Recognition from Biomedical Text Using SVM [2]

Support Vector Machines are used for the NER task by extracting features to conduct identification. Some examples of features used are Part-of-speech Tags, word shape and context words. SVM is a very effective tool that has been widely used for both text and image classification and machine

learning. Here SVM is used to recognize the classification of biomedical named entities to improve efficiency.

Named Entity Recognition in Malayalam using Fuzzy Support Vector Machine [1]

The system presented in this paper is a supervised machine learning approach for achieving NER in a Malayalam text. For the purpose of identifying the named entities, contextual semantic features and linguistic grammar features are used as rules. The system uses support vector machine tool for classification of named entities into predefined classes, like Name, Organization, Place and Date class. SVM solves binary classification problem using a linear classifier but can be extended to a multi classifier. A fuzzy module is also integrated into the system so that ambiguities can be resolved.

3. Challenges faced in Sanskrit

Basic Problems faced in NER

As many words are available in different categories, ambiguity can be a problem,

- 1) Common noun Vs proper noun
E.g. 'Suraj' (name,sun)
- 2) Organization Vs person name
E.g. 'Amulya'(name of a company or person)
- 3) Organization Vs place name
- 4) Person name Vs place name

Problems faced in Sanskrit language

- 1) Lack of capitalization, It follows SOV (Subject-Object-Verb) instead SVO in English. Also the phoneme 'A' and the full stop is denoted by '|'
- 2) Unavailability of gazetteer lists and ambiguity in Indian names, as lists are available in digital format there can be problems faced
- 3) No Trained dataset
- 4) Highly inflectional, agglutinative and morphologically rich in nature
- 5) Multiple representations are available in text formats.

4. Approaches to Named Entity Recognition

NER systems can be implemented using Rule-based or Machine Learning based approach. Rule based approaches uses rules written manually by experts and provides high accuracy compared to alternate approaches whereas Machine learning approaches are fully dependent on arithmetical models to predict Named Entity (NE) in a given document. A few approaches using arithmetic models are given below:

Hidden Markov Model (HMM)

It is an algebraic Markov model where one assumes that the system to be modeled is a Markov process which consists of an unobserved state.

The HMM can be represented as a simple dynamic Bayesian network. In other Markov models like Markov chains, the states are not hidden and can be viewed by the user, hence state transition probabilities are the only parameters, whereas HMMs assume that the observed data is not the actual state of the model, instead it is actually generated by the hidden states. Usually this would make the inference difficult, but the Markov Property that HMMs possess makes the inference efficient.

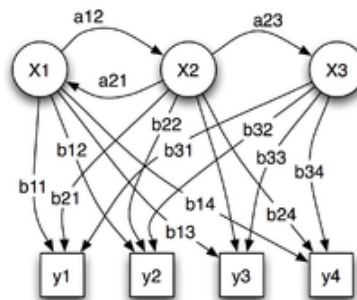


Figure 2: A simple Hidden Markov Model with 3 states and 4 observation tokens. The following labels represent the states x_1, x_2 and x_3 , the transition probabilities is shown by the edge label a_{ij} for the probability of moving from state x_i to state x_j . Observations are represented by y_1, y_2, y_3 and y_4 . The probability of state generating observation is shown by the edge label b_{ij} [15]

Each state has a probability distribution over the possible output tokens. Therefore, the information about the sequence of states can be gained from the sequence of tokens generated by an HMM.

The assumptions of an HMM are the following:

- Every state is dependent of its immediate predecessor.
- Each observation value is dependent on the current state.
- All observations need to be enumerated.

The equation for HMM is given as:

$$P(X) = \sum \prod_{i=0}^n P(y_i(y_i - 1)p(x_i|y_i))$$

Where,

$$X = (x_1 \dots x_n)$$

$$Y = (y_1 \dots y_n)$$

$$P(x) = \text{Probability of observing sequence } X$$

Because of their flexibility and computational efficiency, Hidden Markov Models have found a wide application in many different fields. They are widely known for their use in temporal pattern recognition and generation, such as handwriting recognition and speech synthesis.

Co-Training Algorithm

The idea of co-training is utilizing the unlabelled data to make a better classification system with the use of limited amount of labeled data. Co-training uses two independent views as the learning process.

The assumptions taken into consideration are that, each view alone is sufficient to make a good classification, given enough labeled data and given the class labels, the two views are conditionally independent and are described in the algorithm [14].

Conditional Random Fields (CRF)

CRF is a discriminative undirected and probabilistic graphical model. It is used to encode known relationships between observations and construct consistent interpretations. CRF is a graphical model whose nodes can be divided into exactly two disjoint sets X and Y , the observed and output variables, respectively; the conditional distribution $P(Y/X)$ is then modeled.

There exist special cases where an exact inference is feasible: If the graph is either a tree or a chain, message passing algorithms will produce exact solutions. In such cases, the algorithms used are similar to the forward-backward and Viterbi algorithm for the case of HMMs.

The conditional probability of a state sequence $S = (s_1, s_2, \dots, s_T)$ given an observation sequence $O = (o_1, o_2, o_3, \dots, o_T)$ is calculated as:

$$P(S|O) = \frac{1}{Z_0} \exp \sum_{t=1}^T \sum_k \lambda_k f_k(S_{t-1}, S_t, o, t)$$

Where Z_0 is a normalization factor over all state sequence.

$$Z_0 = \sum \exp \left(\sum_{t=1}^T \sum_k \lambda_k f_k(S_{t-1}, S_t, o, t) \right)$$

And $f_k(S_{t-1}, S_t, o, t)$ is a feature function whose weight λ_k is to be learned via training.

CRF++

CRF++ is a customizable open source implementation of CRFs for segmenting/labeling sequential data. CRF++ is designed for generic purposes and can be applied to tasks like Named Entity Recognition, Information Extraction etc.

Artificial Neural Network(ANN)

The main task of a neural network is to automatically learn to recognize patterns and make smart decisions using the data available, so that it can be applied to classify new texts within large databases. Artificial neural networks (ANNs), a form of connectionism, are computing systems that are inspired by the biological nervous systems of humans and animals. Such systems learn to do tasks by making use of examples, usually without task-specific programming. For example, in image recognition, neural networks can learn to identify images that contain cars by analyzing sample images that are manually labeled as car or no car and use the results to identify cars in other images. Neural networks have found most use in applications where it is difficult to express in a traditional computer algorithm using rule-based programming.

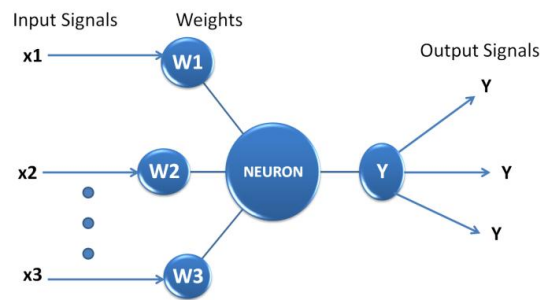


Figure 3: The neuron as a simple computing element [3]

The goal of the neural network method is to solve problems just like a human brain would solve problems.

The four steps in implementing ANN's are:

- 1) Initialization
- 2) Activation
- 3) Weight training
- 4) Iteration

Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyper plane. SVM training algorithms form a non-probabilistic binary linear classifier by building models that will classify new data into one category or another.

It is a supervised machine learning algorithm which can be used in both classification and regression challenges. It is mostly used in classification problems. Here, each data item is plotted as a point in an n -dimensional space, n being the number features that are considered, along with the value of each feature as the value of a specific coordinate. Then, classification is performed by finding the hyper-plane that would most efficiently differentiate the two classes. Support Vectors are the co-ordinates of individual observation.

Basically, an SVM will learn to find a linear hyper-plane that will separate both positive and negative examples with maximum degree of margin. This has proved to have good properties while generalizing bounds for the induced classifiers. The maximal margin can be express as follows:

$$(w \cdot x) + b = 0, (w \in R^n, b \in R)$$

The hyper-plane separates the training data into positive and negative parts, such that:

$$y_i(w \cdot x_i) \geq 1$$

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using a kernel. A kernel maps the inputs into high-dimensional feature spaces which makes multi classification possible.

Kernel Functions:

$$K(X_i, X_j) = \begin{cases} X_i X_j & \text{Linear} \\ (\gamma X_i \cdot X_j + C)^d & \text{Polynomial} \\ \exp(-\gamma |X_i - X_j|^2) & \text{RBF} \end{cases}$$

Where

$K(X_i, X_j) = \phi(X_i) \cdot \phi(X_j)$ and γ is an adjustable parameter of certain kernel functions

The kernel function represents the dot product of the input data points that are mapped into higher dimensional feature space by transformation ϕ .

Supervised learning is not possible on unlabeled data; hence an unsupervised learning approach is needed. The unsupervised method attempts to find natural clustering of the data to groups, and then maps new data to these formed groups.

Fuzzy Support Vector Machine (F-SVM)

A support vector machine learns from the training set to classify data into two distinct classes. In many cases, each input point may not be fully assigned to one of the two classes. This calls for a mechanism which resolves such ambiguities. A fuzzy membership function, when applied to each input point can reformulate the SVMs such that different input points will contribute to the decision surface distinctly. This is called fuzzy SVMs (FSVMs). Fuzzy logic is a type of many-valued logic where the truth values of variables can be any real number between 0 and 1.

SVM causes ambiguity in classifying the named entities to predefined classes. Fuzzy logic module is integrated with the SVM to overcome this weakness. During the testing phase, the SVM classifier is integrated with the fuzzy module. The output of the SVM classifier is analyzed and checked if a named entity belongs to more than one class. When multi-classification occurs the fuzzy module checks for the more accurate class for the ambiguous NE and the assigns the entity to more accurate class.

Fuzzy logic enables values to be more accurate than on or off. While Boolean logic only allows true or false, fuzzy logic allows all things in between. To put it in more precise mathematical terms, classical logic has two values (0 or 1). With fuzzy logic, a value of 0.7 or 0.571 is possible. Fuzzy logic finds its use mostly in expert systems.

5. Accuracy Comparison

From the above survey we have seen that though the work on NER in Sanskrit language is limited, substantial work has been done for similar Indian Languages. The level of accuracy obtained for each method studied is shown using a bar graph. We can see that Fuzzy-SVM is the method with the highest accuracy in comparison to the other approaches.

This survey reveals that 'Named entity recognition in Malayalam using fuzzy support vector machine' has achieved highest accuracy using FSVM 92.8%, using ANN 88%, SVM 84.248%, CRF 74%, Co-Training 73% and using HMM 62.7%. Hence we can arrive at the conclusion that Named Entity recognition in Sanskrit can be best achieved using Fuzzy Support Vector Machine.

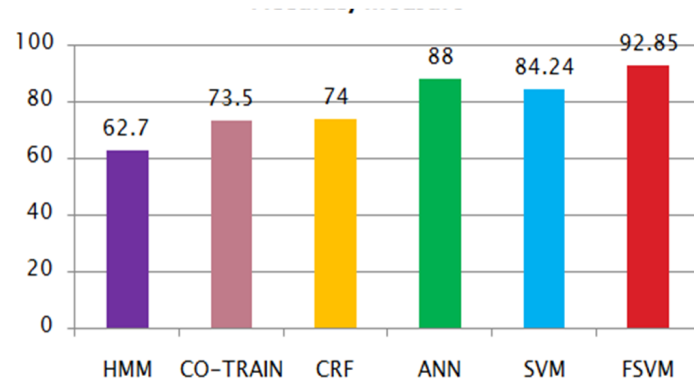


Figure 4: Accuracy Comparison

6. Conclusion

Various methods to perform Named entity recognition are analyzed. It is very well understood that all methods work well for different purposes. Each method has various advantages and disadvantages. The system with the highest accuracy is the Fuzzy Support Vector Machine.

References

- [1] Lakshmi G, JanuR.Panicker, Meera M, 2016Aug. Named entity recognition in Malayalam using fuzzy support vector machine. International Conference on Information Science (ICIS).
- [2] ZhenfeiJu, Jian Wang, Fei Zhu 2011 May, Named Entity Recognition from Biomedical Text Using SVM, 5th International Conference on Bioinformatics and Biomedical Engineering, 2011
- [3] Naji F. Mohammed and NazliaOmar , 2012. Arabic Named Entity Recognition Using Artificial Neural Network .Journal of Computer Science Volume 8, Issue 8, 2012
- [4] M. Bindu and S. M. Idicula, 2011.Named entity recognizer employing multiclass support vector machines for the development of question answering systems.International Journal of Computer Applications, vol. 25, no. 10, pp. 4046.
- [5] A. Mansouri, L. S. Affendy and A. Mamat, 2008.A new fuzzy support vector machine method for named entity recognition.Computer Science and Information Technology, 2008. ICCSIT08. International Conference on, pp. 2428, IEEE.
- [6] K. Srinivasagan, S. Suganthi, and N. Jeyashenbagavalli, 2014. An automated system for tamil named entity recognition using hybrid approach. Intelligent Computing Applications (ICICA), 2014 International Conference on, pp. 435439, IEEE.
- [7] S. B. PrakashHiremath. Approaches to named entity recognition in Indian languages: A study. International Journal of Engineering and Advanced Technology (IJEAT), ISSN, vol. 2249, p. 8958.
- [8] ApurbalalSenapati, Arjun Das, UtpalGarain. Named-Entity Recognition in Bengali , FIRE '12 and '13Post-Proceedings of the 4th and 5th Workshops of the Forum for Information Retrieval EvaluationArticle No. 14,2013
- [9] BayuAryoyudanta, TeguhBharataAdji, IndrianaHidayah. 2016 July.Semi supervised learning approach for Indonesian Named Entity Recognition (NER) using co-training algorithm. International Seminar on Intelligent Technology and Its Applications (ISITIA), July 2016

- [10] KomilVora,AvaniVasant,RachitAdhvaryu. 2012. Named entity recognition and classification for Gujarati language. International Journal of Computing, Communications and Networking.
- [11] Chun-Fu Lin, Sheng-De Wang, 2002. Fuzzy support vector machines, IEEE Transactions on Neural Networks (Volume: 13, Issue: 2), Mar 2002.
- [12] Collins, Michael and Y. Singer. Unsupervised models for named entity classification, In proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.
- [13] PawandeepKaur andAmandeepKaur, 2016, Named entity Recognition Model for Punjabi Language: A Survey in 2nd International Conference on Contemporary Computing and Informatics.
- [14] A. Blum & T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training", page 4, [Online]. Available: <https://profs.info.uaic.ro/~ciortuz/SLIDES/co-training.pdf>. [Accessed Feb 15, 2018].
- [15] John McGonagle, Ayush Rai, and Eli Ross, "Hidden Markov Models", [Online]. Available: <https://brilliant.org/wiki/hidden-markov-models/>. [Accessed Feb. 10, 2018]

Authors Biography



Ms. Joseph Mickole James, Currently doing her postgraduation in Computer Science and Engineering-Information Systems at Rajagiri School of Engineering and Technology, Kochi, India. She completed her graduation in Computer Science and Engineering from Mahatma Gandhi University.



Ms. Sangeetha Jamal, Assistant Professor in the Department of Computer Science & Engineering, Rajagiri School of Engineering & Technology, Cochin, India. She has done her Masters in technology from CUSAT, Kochi. Currently working in the area of NLP and Artificial Intelligence. Has published few papers in the area of Natural Language Processing and has 15 years of teaching experience. Interested in areas like Programming Languages, Natural Language Processing and Artificial

