Domain Focused Named Entity Recognizer for Tamil Using Conditional Random Fields

Vijayakrishna R
AU-KBC Research Centre
MIT Campus, Anna University
Chennai, India
vijayakrishna@au-kbc.org

Sobha L
AU-KBC Research Centre
MIT Campus, Anna University
Chennai, India
sobha@au-kbc.org

Abstract

In this paper, we present a domain focused Tamil Named Entity Recognizer for tourism domain. This method takes care of morphological inflections of named entities (NE). It handles nested tagging of named entities with a hierarchical tagset containing 106 tags. The tagset is designed with focus to tourism domain. We have experimented building Conditional Random Field (CRF) models by training the noun phrases of the training data and it gives encouraging results.

1 Introduction

Named Entity Recognition (NER) is the task of identifying and classifying the entities such as person names, place names, organization names etc, in a given document. Named entities play a major role in information extraction. NER has been a defined subtask in Message Understanding Conference (MUC) since MUC 6. A well performing NER is important for further level of NLP techniques.

In general NER is a hard problem.. Words can have multiple uses and there is an unbounded number of possible names. Many techniques have been applied in Indian and European languages for NER. Some of them are rule based system (Krupka and Hausman, 1998), which makes use of dictionary and patterns of named entities, Decision trees (Karkaletsis et al., 2000), Hidden Morkov Model (HMM) (Biker, 1997), Maximum Entropy Morkov Model (MEMM) (Borthwick et al., 1998), Conditional Random Fields (CRF) (Andrew McCallum and Wei Li, 2003) etc. In short, the approaches can be classified as rule-based approach, machine learning approach or hybrid approach.

For Indian languages, many techniques have been tried by different people. MEMM system for Hindi NER (Kumar and Pushpak, 2006) gave an average F1 measure of 71.9 for a tagset of four named entity tags.

NER has been done generically and also domain specific where a finer tagset is needed to describe the named entities in a domain. Domain specific NER is common and has been in existence for a long time in the Bio-domain (Settles 2004) for identification of protein names, gene names, DNA names etc.

We have developed a domain specific hierarchical tagset consisting of 106 tags for tourism domain. We have used Conditional Random Fields, a machine learning approach to sequence labeling task, which includes NER.

Section 2 gives a brief introduction to Conditional Random Fields (CRF). Section 3 discusses the nature of named entities in Tamil, followed by section 4 describing the tagset used in tourism domain. Section 5 describes how we have presented the training data to build CRF models and how we have handled nested tagging. Sections 6 and 7 explain the experiments and results. The paper is concluded in section 8.

Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, pages 59–66,
Hyderabad, India, January 2008. ©2008 Asian Federation of Natural Language Processing

2 Conditional Random Fields (CRF)

Conditional Random Fields (CRF) (Lafferty et al., 2001) is a machine learning technique. CRF overcomes the difficulties faced in other machine learning techniques like Hidden Markov Model (HMM) (Rabiner, 1989) and Maximum Entropy Markov Model (MEMM) (Berger et al., 1996). HMM does not allow the words in the input sentence to show dependency among each other. MEMM shows a label bias problem because of its stochastic state transition nature. CRF overcomes these problems and performs better than the other two. HMM, MEMM and CRF are suited for sequence labeling task. But only MEMM and CRF allows linguistic rules or conditions to be incorporated into machine learning algorithm.

Lafferty et al, define Conditional Random Fieds as follows**:** “Let **G** = (**V**,**E**) be a graph such that **Y** = $(\mathbf{Y}_v)_{v\in\mathbf{V}}$, 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 $\mathbf{Y}_v$ obey the Markov property with respect to the graph: $p(\mathbf{Y}_v|\mathbf{X},\mathbf{Y}_w,w?v) = p(\mathbf{Y}_v|\mathbf{X},\mathbf{Y}_w,w\sim v)$, where w~v means that w and v are neighbors in G”.

Here **X** denotes a sentence and **Y** denotes the label sequence. The label sequence **y** which maximizes the likelihood probability $p_?(\mathbf{y}|\mathbf{x})$ will be considered as the correct sequence, while testing for new sentence **x** with CRF model **?** . The likelihood probability $p_?(\mathbf{y}|\mathbf{x})$ is expressed as follows.

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

where $?_k$ and μ_k are parameters from CRF model ? and f_k and g_k are the binary feature functions that we need to give for training the CRF model. This is how we integrate linguistic features into machine learning models like CRF.

In NER task, the sequence of words which forms a sentence or a phrase can be considered as the sequence **x** and the sequence formed by named entity label for each word in the sequence **x** is the label sequence **y**. Now, the task of finding **y** that best describes **x** can be found by maximizing the likelihood probability $p_?(\mathbf{y}|\mathbf{x})$. Thus, NER task can be considered as a sequence labeling task. Hence CRF can be used for NER task.

3 Characteristics of Named Entities in Tamil

Unlike English, there is no concept of capital letters in Tamil and hence no capitalization information is available for named entities in Tamil. All named entities are nouns and hence are Noun Phrases. But not all Noun Phrases are Named Entities. Since named entities are noun phrases, they take all morphological inflections. This makes a single named entity to appear as different words in different places. By applying Morphological analysis on words, the root words of inflected Named Entities can be obtained. These roots will be uninflected Named Entities which is what is required in most applications. Some type of named entities like date, money etc, occur in specific patterns.

Example for inflected named entity:

ceVnYnYE**kku** (“to Chennai”).

Example for pattern in named entity:

2006 aktopar 25Am wewi (“25th October, 2006”)

Pattern: <4 digits> <month> <1-2 digit> [Am wewi]

4 Named Entity Tagset used

The tagset which we use here for NER contains 106 tags related to each other hierarchically. This type of tagset is motivated from “ACE English Annotation Guidelines for Entities” developed by Linguistic Data Consortium. The tagset which we use is built in-house with focus to tourism domain.

4.1 Sample Tags

Sample tags from the entire tagset is shown below with their hierarchy.

1. Enamex
    1.1. Person
        1.1.1. Individual
            1.1.1.1. Family Name
            1.1.1.2. Title

1.1.2. Group
1.2. Organization
. . . .
1.3. Location
. . . .
1.4. Facilities
. . . .
1.5. Locomotive
. . . .
1.6. Artifact
. . . .
1.7. Entertainment
. . . .
1.8. Materials
. . . .
1.9. Livthings
. . . .
1.10. Plants
. . . .
1.11. Disease
. . . .
2. Numex
2.1. Distance
2.2. Money
2.3. Quantity
2.4. Count
3. Timex
3.1. Time
3.2. Year
3.3. Month
3.4. Date
3.5. Day
3.6. Period
3.7. Sday

Certain tags in this tagset are designed with focus to Tourism and Health Tourism domain, such as place, address, water bodies (rivers, lakes etc.,), religious places, museums, parks, monuments, airport, railway station, bus station, events, treatments for diseases, distance and date.

The tags are assigned with numbers 1,2,3 for zeroth level, the tags with numbers 1.1, 1.11, 2.1 ,2.4 and 3.1 ,3.7 etc for level-1, the tags with numbers 1.1.1, 1.1.2, 1.2.1 etc as level-2 and the tags with numbers 1.1.1.1, 1.1.1.2, 1.2.4.1 etc for level-3 because they occur in the hierarchy in corresponding levels. We have 3 tags in zeroth level, 22 tags in level-1, 50 tags in level-2 and 31 tags in level-3.

4.2 Sample Annotation

Tamil :

<person> <city> mawurE </city> <individual> manYi <familyname> Eyar </familyname> </individual> </person> <city> ceVnYnYEkku </city> vanwAr.

English equivalent :

<person> <city> Madhurai </city> <individual> Mani <familyname> Iyer </familyname> </individual> </person> came to <city> Chennai </city>.

5 NER using CRF

We used CRF++ (Taku Kudo, 2005), an open source toolkit for linear chain CRF. This tool when presented with the attributes extracted from the training data builds a CRF model with the feature template specified by us. When presented with the model thus obtained and attributes extracted from the test data, CRF tool outputs the test data tagged with the labels that has been learnt.

5.1 Presenting training data

Training data will contain nested tagging of named entities as shown in section 4.2. To handle nested tagging and to avoid ambiguities, we isolate the tagset into three subsets, each of which will contain tags from one level in the hierarchy. Now, the training data itself will be presented to CRF as three sets of training data. From this, we will get three CRF models, one for each level of hierarchy.

Example:

The sample sentence given in section 4.2 will be presented to CRF training for each level of hierarchy as follows:

Level-1:

<location> mawurE </location> <person> manYi Eyar </person> <location> ceVnYnYEkku </location> vanwAr.

Level-2:

<place> mawurE </place> <individual> manYi Eyar </individual> <place> ceVnYnYEkku </place> vanwAr.

Level-3:

<city> mawurE </city> manYi <familyname> Eyar </familyname> <city> ceVnYnYEkku </city> vanwAr.

Notice that the tags ‘location’ and ‘place’ are not specified in the input sentence. In the hierarchy, the ‘location’ tag is the parent tag of ‘place’ tag which is a parent tag of ‘city’ tag. Thus for the word “mawurE”, level-1 tag is ‘location’, level-2 tag is ‘place’ and level-3 tag is ‘city’.

5.2 Attributes and Feature Templates

Attributes are the dependencies from which the system can infer a phrase to be named entity or not. Features are the conditions imposed on these attributes. Feature templates help CRF engine to form features from the attributes of the training data. From the characteristics of named entities in Tamil, we see that it is only the noun phrases that are possible candidates for Named Entities. So we apply Noun Phrase Chunking and consider only noun phrases and train on them. The attributes that we arrived at are explained below:

1. **Roots of words**: This is to ignore inflections in named entities. Also to learn the context in which the named entity occurs, we consider two words prior and two words subsequent to the word under analysis and take unigram, bigram and trigram combinations of them as attributes.
2. **Their Parts of Speech (POS)**: This will give whether a noun is proper noun or common noun. POS of current word is considered.
3. **Words and POS combined**: The present word combined with the POS tag of the previous two words and the present word combined with POS of the next two words are taken as features.
4. **Dictionary of Named Entities**: A list of named entities is collected for each type of named entities. Root words are checked against the dictionary and if present in the dictionary, the dictionary feature for the corresponding type of named entity is considered positive.
5. **Patterns**: Certain types of named entities such as date, time, money etc., show patterns in their occurrences. These patterns are listed out. The current noun phrase is checked against each pattern. The feature is taken as true for those patterns which are satisfied by the current noun phrase.

Example Patterns:

Date: <4 digits> <month> <1-2 digit> [Am wewi]

Money: rU. <digits> [Ayiram|latcam|koti]

(English Equivalent:

Rs. <digits> [thousands|lakhs|crores])

6. Bigram of Named Entity label

A feature considering the bigram occurrences of the named entity labels in the corpus is considered. This is the feature that binds the consecutive named entity labels of a sequence and thus forming linear chain CRFs. Sample noun phrase with level-1 tags:

arulYmiku	JJ	person
cupramaNiya	NNPC	person
cuvAmi	**NNPC**	**person**
wirukoyil	NNC	location
vayalUr	NNP	location

English Equivalent:

Gracious	JJ	person
Subramaniya	NNPC	person
Swami	**NNPC**	**person**
Temple	NNC	location
Vayalore	NNP	location

Attributes are extracted for each token in the noun phrase. For example, the attributes for third token in the sample noun phrase given are as follows.

1. Unigram: arulYmiku, cupramaNiya, cuvAmi, wirukoyil, vayalUr.
2. Bigram: cupramaNiya/cuvAmi, cuvAmi/ wirukoyil
3. Trigram: cupramaNiya/cuvAmi/wirukoyil
4. POS of current word: NNPC
5. Word and previous 2 POS: JJ/NNPC/ cuvAmi
6. Word and next 2 POS: cuvAmi/NNC/NNP
7. Bigram of NE labels: person/person

The CRF training process described above is illustrated in Figure-1.

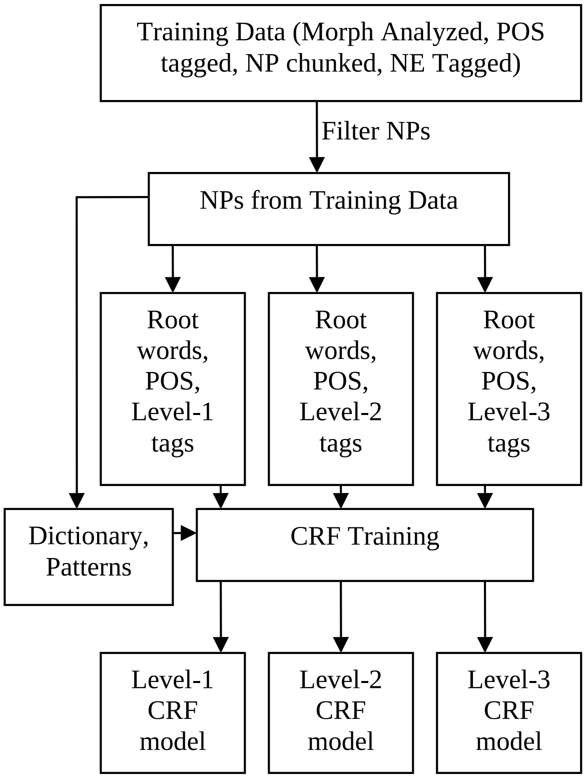
Training Data (Morph Analyzed, POS tagged, NP chunked, NE Tagged)
Filter NPs
NPs from Training Data
Root words, POS, Level-1 tags
Root words, POS, Level-2 tags
Root words, POS, Level-3 tags
Dictionary, Patterns
CRF Training
Level-1 CRF model
Level-2 CRF model
Level-3 CRF model

Figure 1.Training CRF for NER

5.3 Presenting testing data

Test data will also be presented in way similar to how we presented the training data. Test data is processed for Morph analysis, POS (Arulmozhi et al., 2004) and NP chunking (Sobha and Vijay Sundar Ram, 2006). Here also, the same set of attributes and feature templates are used. Now, the test data is tagged with each of the CRF models built for three levels of hierarchy. All the three outputs are merged to get a combined output. The CRF testing is illustrated in Figure 2.

6 Experiments

A 94k words corpus is collected in Tamil for tourism domain. Morph Analysis, POS tagging, NP chunking and named entity annotation are done manually on the corpus. This corpus contains about 20k named entities. This corpus is split into two sets. One forms the training data and the other forms the test data. They consist of 80% and 20% of the total data respectively. CRF is trained with training data and CRF models for each of the levels in the hierarchy are obtained. With these models the test data is tagged and the output is evaluated manually.

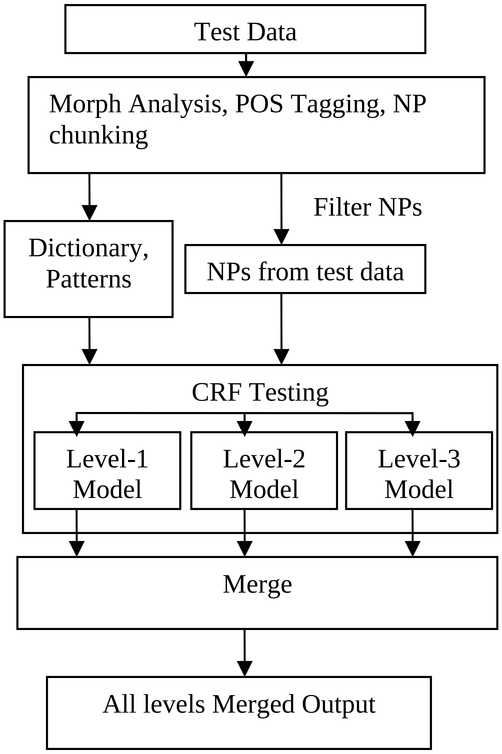

Figure 2. CRF Testing for NER

7 Results

The results of the above experiment are as follows. Here, NE means Named Entity, NP means noun phrase.

Number of NPs in test data = 7922

There are totally 4059 NEs in the test data. All of them bear level-1 tags. Out of 4059 NEs, 3237 NEs bear level-2 tags and 727 NEs bear level-3 tags. The result from the system is shown in Table 1 and Table 2.

The system performs well for domain focused corpus. It identifies inflected named entities efficiently by considering the root form of each word in noun phrases. The reason for good precision is that tagging is done only when the root word that it is seeing is already learnt from the training corpus or the context of the current word is similar to the context of the named entities that it has learnt from the training corpus. However, in some words like 'arccunYAnawi' (Arjuna River), the Morph Analyzer gives two root words which are 'arccunYa' and 'nawi'. For our case, only the first word is considered and the system tags it as 'person' instead of 'waterbodies'.

Named Entity Level	Level-1	Level-2	Level-3
Number of NEs in data	4059	3237	727
Number of NEs identified by NER engine	3414	2667	606
Number of NEs identified correctly	3056	2473	505
Precision %	89.51	**92.73**	83.33
Recall %	75.29	**76.40**	69.46
F1 measure %	81.79	**83.77**	75.77

Table 1. Evaluation of output from NER engine for each level

Performance Measure	Value in %
Precision	88.52
Recall	73.71
F1 Measure	80.44

Table 2. Overall result from NER engine

When there are new named entities which are not in training corpus, CRF tries to capture the context and tags accordingly. In such cases irrelevant context that it may learn while training will cause problem resulting in wrong tagging. This affects the precision to some extent. When the named entities and their context are new to CRF, then they are most likely not tagged. This affects the recall.

From Table 1, we see that the system performs better for level-2 tags than for level-1 tags even though level-1 tags are less in number than level-2 tags and occur more frequently than level-2 tags. This is so because the named entities with level-2 tags have relatively more context and are lesser in length (number of words in the named entity) than the named entities in level-1 tags. Level-3 tags contain lesser number of tags than level-2 tags and also occur less frequently. Because of relatively more data sparseness, the system is unable to perform well for level-3 tags as it can for other levels.

8 Conclusion

We see that Conditional Random Fields is well suited for Named Entity recognition task in Indian languages also, where the inflection of named entities can be handled by considering their root forms. A good precision can be obtained by presenting only the noun phrases for both testing and training.

References

Arulmozhi P, Sobha L and Kumara Shanmugam B. 2004. *Parts of Speech Tagger for Tamil*, Symposium on Indian Morphology, Phonology & Language Engineering, March 19-21, IIT Kharagpur. :55-57.

Berger A, Della Pietra S and Della Pietra V. 1996. *A Maximum Entropy Approach to Natural Language Processing*. Computational Linguistics, 22(1).

Bikel D M. 1997. *Nymble: a high-performance learning name-finder*. In Proceedings of the Fifth Conference on Applied Natural Language Processing. :194-201.

Borthwick A, Sterling J, Agichtein E and Grishman R. 1998. *Description of the MENE named Entity System*, In Proceedings of the Seventh Machine Understanding Conference (MUC-7).

Karkaletsis V, Pailouras G and Spyropoulos C D. 2000. *Learning decision trees for named-entity recognition and classification*. In Proceedings of the ECAI Workshop on Machine Learning for Information Extraction.

Krupka G R and Hausman K. 1998. *Iso Quest Inc: Description of the NetOwl Text Extraction System as used for MUC-7*. In Proceedings of Seventh Machine Understanding Conference (MUC 7).

Kumar N, Pushpak Bhattacharyya. 2006. *Named Entity Recognition in Hindi using MEMM*.

John Lafferty, Andrew McCallum, Fernando Pereira. 2001. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. In Proceedings of the Eighteenth International

Conference on Machine Learning (ICML-2001). 282-289.

Andrew McCallum and Wei Li. 2003. *Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons*. Seventh Conference on Natural Language Learning (CoNLL).

Lawrence R. Rabiner. 1989. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. In Proceedings of the IEEE, 77(2):257–286.

Settles B. (2004). *Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets*. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA), Geneva, Switzerland. pp:104-107.

Sobha L, Vijay Sundar Ram R. 2006. *Noun Phrase Chunking in Tamil*. In proceedings of the MSPIL-06, IIT Bombay. pp:194-198.

Taku Kudo. 2005. *CRF++, an open source toolkit for CRF*, http://crfpp.sourceforge.net .