Named Entity Recognition using Hidden Markov Model (HMM)

Sudha Morwal [1], Nusrat Jahan [2] and Deepti Chopra [3]

[1]Associate Professor, Banasthali University, Jaipur, Rajasthan-302001
sudha_morwal@yahoo.co.in

[2] M.Tech (CS), Banasthali University, Jaipur, Rajasthan-302001
nusratkota@gmail.com

[3] M. Tech (CS), Banasthali University, Jaipur, Rajasthan-302001
deeptichopra11@yahoo.in

Abstract

Named Entity Recognition (NER) is the subtask of Natural Language Processing (NLP) which is the branch of artificial intelligence. It has many applications mainly in machine translation, text to speech synthesis, natural language understanding, Information Extraction, Information retrieval, question answering etc. The aim of NER is to classify words into some predefined categories like location name, person name, organization name, date, time etc. In this paper we describe the Hidden Markov Model (HMM) based approach of machine learning in detail to identify the named entities. The main idea behind the use of HMM model for building NER system is that it is language independent and we can apply this system for any language domain. In our NER system the states are not fixed means it is of dynamic in nature one can use it according to their interest. The corpus used by our NER system is also not domain specific.

Keywords

Named Entity Recognition (NER), Natural Language processing (NLP), Hidden Markov Model (HMM).

1.Introduction

Named Entity Recognition is a subtask of Information extraction whose aim is to classify text from a document or corpus into some predefined categories like person name, location name, organisation name, month, date, time etc. And other to the text which is not named entities. NER has many applications in NLP. Some of the applications include machine translation, more accurate internet search engines, automatic indexing of documents, automatic question-answering, information retrieval etc. An accurate NER system is needed for these applications.
Most NER systems use a rule based approach or statistical machine learning approach or a Combination of these. A Rule-based NER system uses hand-written rules frame by linguist which are certain language dependent rules that help in the identification of Named Entities in a document. Rule based systems are usually best performing system but suffers some limitation such as language dependent, difficult to adapt changes.

Machine-learning (ML) approach Learn rules from annotated corpora. Now a day's machine learning approach is commonly used for NER because training is easy, same ML system can be used for different domains and languages and their maintenance is also less expensive? There are various machine learning approaches for NER such as CRF (conditional Random Fields),

MEMM (Maximum Entropy Markov Model), SVM (Support Vector Machine) and HMM (Hidden Markov Model) and dictionary based approach. Among all these HMM, being the most promising, has not been explored in its full potential for NER. The work that has been reported is domain specific and does not establish it as a general technique.

Mostly the researcher uses hybrid NER system which take advantages of both rule-based and statistical approaches so that the performance of NER system can be improved.

2.Challenges of NER in Indian Languages

In technology for Indian languages NER has an essential need. NER in Indian Languages is a more challenging problem as compared to languages using Roman script due to absence of capitalization, resources etc. Because of these issues any English NER system cannot be used directly for performing NER for Indian language.

To get various features we adapt Hidden Markov Model machine learning approach for Named Entity Recognition in Indian language. Which can be used as general techniques?

- For English and other European languages, capitalization plays a very important role to identify NEs but for Indian languages there is no concept of capitalization which makes NER difficult for these languages.

- Large number of ambiguity exists in Indian names and this makes the recognition a very difficult task for Indian language.

- Indian languages are also a resource poor language. Annotated corpora, name dictionaries, good morphological analyzers, POS taggers web source for name list etc are not yet available in the required quantity and quality [2].

- Lack of standardization and spelling [3].

- Although Indian languages have a very old and rich literary history still technology development are recent [2].

- India is a multilingual country with different language and there is large number of variation in each language. Because of these variations Named entity recognition systems for one language domain do not usually work well in other language domains.

- Indian languages are relatively free-order languages [2].

3.Our approach

3.1 Hidden Markov Model based machine learning

HMM stands for Hidden Markov Model. HMM is a generative model. The model assigns the joint probability to paired observation and label sequence [6]. Then the parameters are trained to maximize the joint likelihood of training sets [6].

Among all approaches, the evaluation performance of HMM is higher than those of others [7]. The main reason may be due to its better ability of capturing the locality of phenomena, which indicates names in text [7].

We can define HMM in a formal way as follows:

$\lambda = (A, B, \pi)$. Here, A represents the transition probability. B represents emission probability and π represents the start probability [4].

A = aij = (Number of transitions from state si to sj) / (Number of transitions from state si) [4].

B = bj (k) = (Number of times in state j and observing symbol k) / (expected number of times in state j) [4].

It means that the word occurs first in a sentence. Baum Welch Algorithm is used to find the maximum likelihood and posterior mode estimates for the HMM parameters [9]. Forward Backward Algorithm is used to find the posterior marginal's of all hidden state variables given a sequence of observations/emissions [8].

3.2. Viterbi algorithm

The Viterbi algorithm (Viterbi 1967) is implemented to find the most likely tag sequence in the state space of the possible tag distribution based on the state transition probabilities [10]. The Viterbi algorithm allows us to find the optimal tags *in* linear time. The idea behind the algorithm is that of all the state sequences, only the most probable of these sequences need to be considered.

Moreover, HMM seems more and more used in NE recognition because of the efficiency of the Viterbi algorithm [Viterbi67] used in decoding the NE-class state sequence [7].

Parameters of HMM Viterbi algorithm is following:

Set of States, S where |S|=N. Here, N is the total number of states.
Observations O where |O|=k .Here, k is the number of Output Alphabets.
Transition Probability, A
Emission Probability B
Initial State Probabilities π

HMM may be represented as: $\lambda = (A, B, \pi)$ [4].

3.3 Current NER in Indian Language

Current work in Indian language regarding NER suffer from following limitations

- Language dependent – NER in one language may not use for other language in any case if it is too much effort required.
- Domain Specific – NER system work best for one domain but in other domain performance is not up to the mark.
- The rule based method gives high accuracy up to certain extent but it requires language experts to construct rule for any language domain.

- NER process requires much time and effort.

- The accuracy of Gazetteer method is acceptable but it has problem when corpus is very large. Since the Indian languages are free format languages and new words are generated rapidly. So managing the list size is big task [5].

- Gazetteer method also takes lots of time to search any named entities in the list and for each word we have to search the entire list from the beginning [5].

- The problem with Maximum entropy model is that it does not solve the label biasing problem [1].

3.4 HMM based NER

- We can develop NER system which is language independent. They are not specific for particular language domain. We can use it for any language domain
  .
- The HMM based NER system is easily understandable and is easy to implement and analyse. It can be used for any amount of data so the system is scalable.

- It solves Sequence labelling problem very efficiently.

- The states used in the model are also not fixed. One can use it according to their requirements or interest means it is of dynamic in nature.

- The HMM based NER system does not require language experts means if a person has little knowledge about the language in which he/she wants to find named entities can easily run/operate this system.

3.5 Proposed System

Proposed System uses learning by example methodology. It provides easy to use method with minimum efforts for Named Entity Recognition in any natural language. Person has to just annotate his corpus and test the system for any sentence. Steps to be followed for any language are as follows-

1. Data preparation
2. Parameter Estimation(Training)
3. Test the system

3.5.1. Step 1: Data Preparation

We need to convert the raw data into trainable form, so as to make it suitable to be used in the Hidden Markov model framework for all the languages. The training data may be collected from any source like from open source, tourism corpus or simply a plaintext file containing some sentences. So in order to make these file in trainable form we have to perform following steps:

Input : Raw text file
Output: Annotated Text (tagged text)

Algorithm

Step1: Separate each word in the sentence.
Step2: Tokenize the words.
Step3: Perform chunking if required.
Step5: Tag (Named Entity tag) the words by using your experience.
Step6: Now the corpus is in trainable form.

3.5.2. Step 2: HMM Parameter Estimation

Input: Annotated tagged corpus
Output: HMM parameters

Procedure:
Step1: Find states.
Step2: Calculate Start probability (π).
Step3: Calculate transition probability (A)
Step4: Calculate emission probability (B)

3.5.2.1. Procedure to find states

State is vector contains all the named entity tags candidate interested.

Input: Annotated text file
Output: State Vector

Algorithm:
For each tag in annotated text file
If it is already in state vector
Ignore it
Otherwise
Add to state vector

3.5.2.2. Procedure to find Start probability

Start probability is the probability that the sentence start with particular tag.

So start probabilities (π) = $\frac{\text{(Number of sentences start with particular tag)}}{\text{(Total number of sentences in corpus)}}$

Input: Annotated Text file:
Output: Start Probability Vector

Algorithm:
For each starting tag
Find frequency of that tag as starting tag
Calculate π

3.5.2.3. Procedure to find Transition probability

If there is two pair of tags called Ti and Tj then transition probability is the probability of occurring of tag Tj after Ti.

So Transition Probability (A) = $\frac{\text{(Total number ofsequences from Ti to Tj)}}{\text{(Total number of Ti)}}$

Input: annotated text file
Output: Transition Probability

Algorithm:

```
For each tag in states (Ti)
    For each other tag in states (Tj)
        If Ti not equal to Tj
            Find frequency of tag sequence Ti Tj i.e. Tj after Ti
     Calculate A = frequency (Ti Tj) / frequency (Ti)
```

3.5.2.4. Procedure to find emission probability

Emission probability is the probability of assigning particular tag to the word in the corpus or document.

So emission probability (B) = $\frac{\text{(Total number of occurance of word as a tag)}}{\text{(Total occurance of that tag)}}$

I**nput**: Annotated Text file
Output: Emission Probability matrix
Algorithm: For each unique word Wi in annotated corpus
Find frequency of word Wi as a particular tag Ti
Divide frequency by frequency of that tag Ti

3.5.3 Step 3: Testing

After calculating all these parameters we apply these parameters to Viterbi algorithm and testing sentence as an observation to find named entities.

4. Example

Consider these raw text containing 6 sentences of Hindi, Urdu and Punjabi language.

पूण प्रातबध हटाओ : इराक ।
बेनजीर का सुनवाई स्थागत ।
وہ بات کرنے سے انکار کرتے ہیں
میں کبھی کبھار ایک کتاب پڑھتا ہوں
ਵਾਫ਼ਰ ਜਿਆ ।
□□ □□ □□□□ □□

Now the annotated text is as follows:

पूण/OTHER प्रातबध/OTHER हटाओ/OTHER :/OTHER इराक/LOC ।/OTHER

बेनजीर/PER का/OTHER सुनवाई/OTHER स्थागत/OTHER ।/OTHER

OTHER/ہیں OTHER/کرتے OTHER/انکار OTHER/ سOTHER/کرنے OTHER/بات OTHER/وہ
OTHER/ہوں. OTHER/پڑھتا OTHER/کتاب OTHER/ایک OTHER/کبھار OTHER/کبھی OTHER/میں

□□□□/OTHER □□□/OTHER ।/OTHER

□□□/OTHER □□□/OTHER □□□□□□/LOC ।/OTHER

Now we calculate all the parameters of HMM model. These are

States= {OTHER, LOC, PER,}
Start probability (π) =

PER	LOC	OTHER
1/6=0.167	0/6=0.000	5/6=0.833

Table1: Start Probability π

Now Transition probability (A) =

	PER	LOC	OTHER
PER	0	0	1/1=1
LOC	0	0	2/2=1
OTHER	0	2/29=0.069	21/29=0.724

Table2: Transition probability A

Emission Probability (B) = since in the emission probability we have to consider all the words in the file. But it's not possible to display all the words in the table so we just gave the snapshot of first sentence of the file. Similarly we can find the emission probability of all the words.

	पूण	प्रातबध	हटाओ	:	इराक	।
PER	0	0	0	0	0	0
LOC	0	0	0	0	½=0.5	0
OTHER	1/29=0.034	1/29=0.034	1/29=0.034	1/29=0.034	0	4/29=0.138

Table3: Emission probability B.

Testing: The testing sentences are:

पूण प्रातबध स्थागत ।

وہ کبھی کبھار پڑھتا ہوں

□□□□ □□□□ ।

The output of these sentences after testing is:
{'OTHER', 'OTHER', 'OTHER', 'OTHER'}
{'OTHER', 'OTHER', 'OTHER', 'OTHER','OTHER'}
{'LOC','OTHER','OTHER'}

5. FEATURES OF PROPOSED SYSTEM

Our Hidden Markov model based NER system has been trained and tested with different Indian languages namely Hindi, Urdu, and Punjabi etc. We have performing training and testing on our tourism corpus and it gives better performance. The works reported in this paper differ from other previous work in terms of the following points:

- **language independent** –
  This methodology works for any natural language European language also. This work tested for Hindi, Urdu and Punjabi and English.

- **General Approach** –
  This approach is not domain specific. This work tested for tourism corpus, general sentences and stories.

- **High Accuracy** –
  If rich corpus is developed if perform best. During testing we also get accuracy till 90 %.

- **Dynamic** –
  All the parameters used by our system are of dynamic in nature means one can use according to their interest. This work is tested for Person, Location, river Country tags in tourism corpus and Person, time, month, dry-fruits, food items tags in story corpus.

- **Usefulness to other classification** –
  Since the parameters are of dynamic in nature the same NER system can be used for other NLP classification like Part-of-speech tagging etc.

- **Fine grained tagging** –
  Mostly systems allot location tag to name of place, river, palace etc. In this system you can set subclass of location tags according to your need. This system has been tested for country, river, tree etc. tags.

- **Use of Annotated corpus** –
  To use this system you have to design tagged corpus either with the help of proposed system or with other tools. This tagged corpus can be used in other natural language processing applications.

6. CONCLUSION

Building a NER based system in Hindi using HMM is a very conducive and helpful in many significant applications. We have studied various approaches of NER and compared these approaches on the basis of their accuracies. India is a multilingual country. It has 22 Indian Languages. So, there is lot of scope in NER in Indian languages. Once, this NER based system with high accuracy is build, then this will give way to NER in all the Indian Languages and further an efficient language independent based approach can be used to perform NER on a single system for all the Indian Languages. So NER system based on HMM model is very efficient especially for Indian languages where large variation occurs.

7. REFERENCES

[1] Pramod Kumar Gupta, Sunita Arora "An Approach for Named Entity Recognition System for Hindi: An Experimental Study" in Proceedings of ASCNT – 2009, CDAC, Noida, India, pp. 103 – 108.

[2] Shilpi Srivastava, Mukund Sanglikar & D.C Kothari. "Named Entity Recognition System for Hindi Language: A Hybrid Approach" International Journal of Computational Linguistics (IJCL), Volume(2):Issue(1):2011.Availableat: http://cscjournals.org/csc/manuscript/Journals/IJCL/volume2/Issue1/IJCL-19.pdf

[3] "Padmaja Sharma, Utpal Sharma, Jugal Kalita"Named Entity Recognition: A Survey for the Indian Languages"(Language in India www.languageinindia.com 11:5 May 2011 Special Volume: Problems of Parsing in Indian Languages.) Available at: http://www.languageinindia.com/may2011/padmajautpaljugal.pdf.

[4] Lawrence R. Rabiner, " A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", In Proceedings of the IEEE, VOL.77,NO.2, February 1989.Available at: http://www.cs.ubc.ca/~murphyk/Bayes/rabiner.pdf.

[5] Sujan Kumar Saha, Sudeshna Sarkar, Pabitra Mitra "Gazetteer Preparation for Named Entity Recognition in Indian Languages" in the Proceeding of the 6th Workshop on Asian Language Resources, 2008 . Available at: http://www.aclweb.org/anthology-new/I/I08/I08-7002.pdf

[6] B. Sasidhar#1, P. M. Yohan*2, Dr. A. Vinaya Babu3, Dr. A. Govardhan4" A Survey on Named Entity Recognition in Indian Languages with particular reference to Telugu" in IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011 available at : http://www.ijcsi.org/papers/IJCSI-8-2-438-443.pdf.

[7] GuoDong Zhou Jian Su," Named Entity Recognition using an HMM-based Chunk Tagger" in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 473-480.

[8] http://en.wikipedia.org/wiki/Forward–backward_algorithm

[9] http://en.wikipedia.org/wiki/Baum-Welch_algorithm.

[10] Dan Shen, jie Zhang, Guodong Zhou,Jian Su, Chew-Lim Tan" Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain" available at: http://acl.ldc.upenn.edu/W/W03/W03-1307.pdf.

Authors

Sudha Morwal is an active researcher in the field of Natural Language Processing. Currently working as Associate Professor in the Department of Computer Science at Banasthali University (Rajasthan), India. She has done M.Tech (Computer Science), NET, M.Sc (Computer Science) and her PhD is in progress from Banasthali University (Rajasthan), India.

Nusrat Jahan received B.Tech degree in Computer Science and Engineering from R.N. Modi Engineering College, Kota, Rajasthan in 2010.Currently she is pursuing her M.Tech degree in Computer Science and Engineering from Banasthali University, Rajasthan. Her subject of interests includes Natural Language Processing and Information retrieval.

Deepti Chopra received B. Tech degree in Computer Science and Engineering from Rajasthan College of Engineering for Women, Jaipur, Rajasthan in 2011.Currently she is pursuing her M.Tech.degree in Computer Science and Engineering from Banasthali University, Rajasthan. Her subject of research includes Natural Language Processing.