Gazetteer Preparation for Named Entity Recognition in Indian Languages

Sujan Kumar Saha
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
sujan.kr.saha@gmail.com

Sudeshna Sarkar
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
shudeshna@gmail.com

Pabitra Mitra
Indian Institute of Technology
Kharagpur, West Bengal
India - 721302
pabitra@gmail.com

Abstract

This paper describes our approaches for the preparation of gazetteers for named entity recognition (NER) in Indian languages. We have described two methodologies for the preparation of gazetteers[1]. Since the relevant gazetteer lists are more easily available in English we have used a transliteration based approach to convert available English name lists to Indian languages. The second approach is a context pattern induction based domain specific gazetteer preparation. This approach uses a domain specific raw corpus and a few seed entities to learn context patterns and then the corresponding name lists are generated by using bootstrapping.

1 Introduction

Named entity recognition involves locating and classifying the names in text. NER is an important task, having applications in information extraction (IE), question answering (QA), machine translation and in most other NLP applications.

NER systems have been developed for resource-rich languages like English with very high accuracies. But constructing an NER for a resource-poor language is very challenging due to unavailability of proper resources. Name-dictionaries or gazetteers are very helpful NER resources and in most Indian languages there is no reasonable size publicly available list. The web contains lots of such resources, which can be used for Indian language NER development. But most of the web resources are in English. Our approach is to transliterate the relevant English resources and name dictionaries into Indian languages to make them useful for Indian language NER task. But direct transliteration from English to an Indian languages is not easy. Few attempts are taken to build English to Indian language transliteration systems but the word agreement ratio (WAR) reached is upto 69.3% (Ekbal et al., 2006).

We have attempted to build a transliteration system which uses an intermediate alphabet. Both the English and the Indian language strings are transliterated to the intermediate alphabet and for a English-Indian language pair, if the transliterated intermediate alphabet strings are same then we have concluded that the strings are the transliteration of one another. We have transliterated the available English name lists into the intermediate alphabet and these might be used as gazetteers. The Indian language words need to be transliterated to the intermediate format to check whether the word is in a gazetteer or not. This system does not transliterate the English name lists into Indian languages but makes them useful in Indian languages NER task.

Transliteration based approaches are useful when there is availability of English name lists. But when relevant English name lists are not available then also we can prepare gazetteers from raw corpus. We have defined a semi-automatic context pattern (CP) extraction based gazetteer preparation framework. This framework uses bootstrapping to prepare

[1] Specialized list of names for a particular class of Named Entity (NE). For Example, $India$ is in the location gazetteer, $Sachin$ is in the person first name gazetteer.

the gazetteers from a large raw corpus starting from few seed entities. Firstly fixed length patterns are formed using the surrounding words of the seeds. Depending on the pattern precision, the patterns are discarded or generalized by dropping tokens from the patterns. This set of high precision patterns extracts other named entities (NEs) which are added to the seed list for the next iteration of the process. Finally we are able to prepare the required gazetteers. To prove the effectiveness of the gazetteer preparation approach, we have prepared some gazetteers like names of cricketers, names of tennis players etc. from a raw Hindi sports domain corpus. The details of the approaches are given in the following sections.

The paper is organized as follows. Usefulness of gazetteers in NER, transliteration approaches in general and specific for Indian languages and general pattern extraction methodologies are discussed in section 2. Section 3 presents the architecture of the 2-phase transliteration system and preparation of gazetteers using that. In section 4 context pattern extraction based gazetteer preparation is discussed. Finally section 5 concludes the paper.

2 Previous Work

The main approaches to NER are Linguistic approaches and Machine Learning (ML) based approaches. The linguistic approach typically uses rule-based models manually written by linguists. ML based techniques make use of a large amount of annotated training data to acquire high-level language knowledge. Several ML techniques like Hidden Markov Model (HMM)(Bikel et al., 1997), Maximum Entropy Model(MaxEnt) (Borthwick, 1999), Conditional Random Field(CRF) (Li and McCallum, 2004) etc. have been successfully used for the NER task. Both the approaches may make use of gazetteer information to build systems. There are many systems which use gazetteers to improve the accuracy.

Ralph Grishman has developed a rule-based NER system which uses some specialized name dictionaries including names of all countries, names of major cities, names of companies, common first names etc (Grishman, 1995). Another rule based NER system is developed by Wakao et al. (1996) which has used several gazetteers like organization names, location names, person names, human titles etc.

We will now mention some ML based systems. $MENE$ is a MaxEnt based system developed by Borthwick. This system has used 8 dictionaries (Borthwick, 1999), which are: First names (*1,245*), Corporate names (*10,300*), Corporate names without suffix (*10,300*), Colleges and Universities (*1,225*), Corporate suffixes (*244*), Date and Time (*51*) etc. The italics numbers in bracket indicates the size of the dictionaries. The hybrid system developed by Srihari et al.(2000) combines several modules built by using MaxEnt, HMM and handcrafted rules. This system uses the following gazetteers: First name (*8,000*), Family name (*14,000*) and a big gazetteer of Locations (*250,000*). There are many other systems which have used name dictionaries to improve the accuracy. Kozareva (2006) described a methodology to generate gazetteer lists automatically for Spanish and to build NER system with labeled and unlabeled data. The location gazetteer is built by finding location patterns which looks for specific prepositions. And the person gazetteer is constructed with graph exploration algorithm.

Transliteration is also a very important topic and lots of transliteration systems for different languages have been developed using different approaches. The basic approaches for transliteration are phoneme based or spelling-based. A phoneme-based statistical transliteration system from Arabic to English was developed by Knight and Graehl(1998). This system uses a finite state transducer that implements transformation rules to do back-transliteration. A spelling-based model that directly maps English letter sequences into Arabic letters was developed by Al-Onaizan and Knight(2002). Several transliteration systems exist for English-Japanese, English-Chinese, English-Spanish and many other languages to English. But very few attempts have been reported on the development of transliteration systems between Indian languages and English. We can mention a transliteration system for Bengali-English transliteration developed by Ekbal et al.(2006). They have proposed different models modifying the joint source channel model. In that system a Bengali string is divided into transliteration units containing a vowel modifier or $matra$ at the end of each unit. Similarly the English string is also divided into units. Then they defined various unigram, bigram or trigram models depending on the consideration of the contexts of the units. They have also considered linguistic knowledge in the form of possible conjuncts and diphthongs in Bengali and their representations in English. This system is capable of transliterating mainly person names. The highest transliteration accuracy achieved by them is 69.3% Word Agreement Ratio (WAR) for Bengali to English and 67.9% WAR for English to Bengali transliteration.

In the field of IE, patterns play a key role in identifying relevant pieces of information. Soderland et al.(1995), Rillof and Jones(1999), Lin et al.(2003), Downey et al.(2004), Etzioni et al.(2005) described different approaches to context pattern induction. Talukder et al.(2006) combined grammatical and statistical techniques to create high precision patterns specific for NE extraction. An approach to lexical pattern learning for Indian languages is described by Ekbal and Bandopadhyay (2007). They used seed data and annotated corpus to find the patterns for NER.

3 Transliteration based Gazetteer Preparation

Gazetteers or name dictionaries are helpful in NER. We have already discussed about some English NER systems where the usefulness of the gazetteers have been established. However while developing NER systems in Indian languages, we tried to find relevant gazetteers. But we could not obtain openly available gazetteer lists for these languages. But we found that there are a lot of resources of names of Indian persons, Indian places, organizations etc. in English available in the web. In Table 1 we have mentioned some of the sources which contains relevant name lists.

But it is not possible to use the available name lists directly in the Indian language NER task as these are in English. We have decided to transliterate the English lists into Indian languages to make them useful in the Indian language NER task.

List	Web Sources
First Name	http://www.bsnl.co.in/ onlinedirectory.htm http://web1.mtnl.net.in/ directory/ http://www.eci.gov.in/ http://hiren.info/indian-baby-names/ http://www.indiaexpress.com/ specials/babynames/
Surname	http://surnamedirectory.com/ surname-index.html http://web1.mtnl.net.in/ directory/ http://en.wikipedia.org
India Location	http://indiavilas.com/ indiainfo/pincodes.asp http://www.indiapost.gov.in http://www.eci.gov.in/
World Location	http://www.maxmind.com/ app/worldcities http://en.wikipedia.org/wiki

Table 1: Web sources for some relevant name lists

3.1 Transliteration

The transliteration from English to Hindi is quite difficult. English alphabet contains 26 characters whereas the Hindi alphabet contains 52 characters. So the mapping is not trivial. We have already mentioned that for Bengali a transliteration system was developed by Ekbal et al. Similar approach can be used to develop transliteration systems for other Indian languages. But this approach uses a bilingual transliteration corpus, which requires much efforts to built, is unavailable in proper size in all Indian languages. Also using this approach the word agreement ratio obtained is below 70%.

To make the transliteration process easier and more accurate, we have decided to build a 2-phase transliteration module. Our goal is to make decision that a particular Indian language string is in an English gazetteer or not. We need not transliterate directly from Indian language strings to English or English name lists into Indian languages. Our idea is to define an intermediate alphabet and both English and Indian language strings will be transliterated to the intermediate alphabet. For two English-Hindi string pair, if the intermediate alphabet is same then we can conclude that one string is the transliteration of the other.

First of all we need to decide the size of the intermediate alphabet. Preserving the phonetic properties we have defined our intermediate alphabet consisting of 34 characters. To indicate these 34 characters, we have given unique character-id to each character.

3.2 English to Intermediate Alphabet Transliteration

For transliterating English strings into the intermediate state, we have built a phonetic map table. This phonetic map table maps an English n-gram into an intermediate character. A part of the map table is given in Table 2. In the map table, the mapping is from strings of varying length in the English to one character in the intermediate alphabet. In our table the length of the left hand side varies from 1 to 3.

English	Intermediate
A	â
EE, I, II	î
OO, U	û
B, W	b̂
BH, V	v̂
CH	ĉ
R, RH	r̂
SH, S	ŝ

Table 2: A part of the map table

In the following we have described the procedure of transliteration.

Procedure 1: Transliteration
Source string - English, Output string - Intermediate.

1. Scan the source string (S) from left to right.

2. Extract the first n-gram (G) from the string. ($n = 3$)

3. Find if it is in the map-table.

4. If yes, insert its corresponding intermediate state entity into target string T.
   Remove the n-gram from S.
   $S = S - G$.
   Go to step 2.

5. Else set $n = n - 1$.
   Go to step 3.

Here we can take an Indian language name, ‘surabhi’, as example to explain the procedure in details. When the name is written in English, it may be written in several ways like ‘suravi, ‘shuravi’, ‘surabhee’, ‘shurabhi’ etc. The English string ‘surabhi’ is transliterated to ‘ŝûr̂âv̂î’ by the transliterator. Again if we see the transliteration for ‘shuravi’, then also the intermediate transliterated string is same as the previous one.

3.3 Indian Language to Intermediate Alphabet Transliteration

This is a 2-phase process. The first phase transliterates the Indian language string into itrans. Itrans is representation of Indian language alphabets in terms of ASCII. Since Indian text is composed of syllabic units rather than individual alphabetic letters, itrans uses combinations of two or more letters of English alphabet to represent an Indian language syllable. However, there being multiple sounds in Indian languages corresponding to the same English letter, not all Indian syllables can be represented by logical combinations of English alphabet. Hence, itrans uses some non-alphabetic special characters also in some of the syllables. A map table[2], with some heuristic knowledge, is used for the transliteration. For example, the Hindi word ‘surabhi’ is converted ‘sUrabhI’ in itrans.

In the second phase the itrans string is transliterated into the intermediate state using the similar procedure described section 3.2. Here also we use a map-table containing the mappings from itrans to intermediate alphabet. This procedure transliterates the example itrans word ‘sUrabhI’ to ‘ŝûr̂âv̂î’.

3.4 Evaluation

In section 3.2 and 3.3 we have described two phase transliteration with an example word. We have shown that our transliteration system transliterates the Indian language name ‘surabhi’ and the corresponding English strings into the same intermediate

[2] The map table is available at www.aczoom.com/itrans.

string. The system has limitations like sometimes two different strings can be mapped into a same intermediate alphabet string.

For the evaluation of the system we have applied the transliteration system for two languages - Hindi and Bengali. For evaluating the system for Hindi we have created a bi-lingual corpus containing 1070 English-Hindi word pair most of which are names. 980 of them are transliterated correctly by the system. The system accuracy is $980 \times 100/1070 = 91.59$%. For evaluating the system for Bengali, we have used a similar bi-lingual corpus and the system transliterates with 89.3% accuracy.

3.5 Prepared Gazetteer Lists

Previously we have mentioned the web sources where some name lists are available. Names of a particular category are collected from different sources and merged to build a English name list of that category. Then we have applied our transliteration procedure on the list and transliterated the list into the intermediate alphabet. This intermediate alphabet list acts as a gazetteer in NER task in Indian languages. When an Indian language NER system needs to access the gazetteer lists, it transliterates the Indian language strings into the intermediate alphabet, and searches into the list. In the following we have described the prepared gazetteer lists which are useful for a general domain Indian language NER system.

First Name List: This list contains 10,200 first names collected from the web. Most of the collected first names are of Indian origin. Apart from the Indian names, we have also collected some non-Indian names. These non-Indian names are generally the names of some famous persons, like sports stars, film stars, scientists, politicians, who are likely to come in Indian language texts. In our first name list 1500 such names are included.

Surname List: This is a very important list which contains common surnames. We have prepared the surname list from different sources containing about 1500 Indian surnames and 400 other surnames.

Indian Locations: This list contains about 14,000 entities. The names of states, cities and towns, districts, important places in different cities and even lots of village names are collected in the list. The list needs to be processed into a list of unigrams (e.g., *kolakAtA*[3] (Kolkata), *bihAra* (Bihar)), bigrams (e.g., *nayI dillI* (New Delhi), *pashchima bA.nglA* (West Bengal)) and trigrams (e.g. *uttaara chabisha paraganA* (North 24 Pargana)). The words are matched with unigrams, sequences of two consecutive words are matched against bigrams and sequences of three consecutive words are matched against trigrams.

World Location: The list contains the names of the countries, different state and city names in world and also the names of important rivers, mountains etc. The list contains about 4,000 location names. Similar to the Indian location list, this list also needs to be processed as unigram, bigrams and trigrams.

4 Context Pattern Extraction based Gazetteer Preparation

Gazetteers can also be prepared by extracting context patterns. Transliteration based gazetteer preparation is applicable while there is availability of English or parallel language name list. But if such relevant name lists are not available, but a large raw corpus is available, then we can use the context pattern extraction based methodology to prepare the gazetteers. This method seeks some high precision context patterns by using some seed entities and hits the patterns to the raw corpus to prepare the gazetteers.

The overall methodology of extracting context patterns from a raw corpus is summarized as follows:

1. Find a large raw corpus and some seed entities (E) for each class of NEs.
2. For each seed entity e in E, from the corpus find context string(C) comprised of n tokens before e, a placeholder for the class instance and n tokens after e. [We have used $n = 3$]
   This set of words form initial pattern.
3. Search the pattern to the corpus and find the coverage and precision.
4. Discard the patterns having low precision.
5. Generalize the patterns by dropping one or more tokens to increase coverage.

[3] The Indian languages strings are written in italics font and using itrans transliteration.

6. Find best patterns having good precision and coverage.

The details of the context pattern extraction based gazetteer preparation methodology is described in the following subsections. We have taken a Hindi sports domain raw corpus and prepared some gazetteers like names of cricketers, names of tennis players to prove the effectiveness of the proposed methodology.

4.1 Selection of Seed Entity

Context pattern extraction based gazetteer preparation methodology is applied to a sports domain corpus which contains about 20 lakhs words collected from the popular Hindi newspaper "Dainik Jagaran". We have worked on preparing the lists of cricket players, list of tennis players. We have collected the most frequent names to build the seed list. For preparing the list of tennis players, we have taken 5 names as seed entities : Andre Agassi, Steffi Graf, Serena Williams, Roger Federer and Justine Henin. Similarly the seed list of cricket players name list contains only 3 names: Sachin Tendulkar, Brian Lara and Glenn McGrath.

4.2 Context Extraction

To extract the patterns for a particular category, we select a part of the corpus where the target seeds will be available with high frequency. For example to get the patterns for the names of the cricketers, we select a part of the corpus where most of the sentences are cricket related. To select the cricket related sentences, we prepared a list containing the most frequent words related to cricket like, $rAna$ (run), $ballebAja$ (batsman), $gedabAja$ (bowler) etc. Depending upon the presence of such words we have selected the 'part'. In our development the cricket 'part' contains 120K words. Similar 'part' is developed for other gazetteers. For a particular seed, we find the occurrences of the seed entity in the corresponding raw 'part' corpus. Then we extracted three tokens immediately preceding the seed and three tokens immediately following the seed. A placeholder ($CRIC$ for cricketers, $TENS$ for tennis players) replaces the seed. The placeholder and the surrounding tokens t_{-3} t_{-2} t_{-1} $placeholder$ t_{+1} t_{+2} t_{+3}) form the initial set of patterns.

For the seed *sachina tedulkara* (Sachin Tendulkar) we extract 92 initial patterns. Some of which are:

- *ki mAsTara blAsTara* CRIC *ko Tima ke*
- *dravi.Da aura* CRIC *241 nAbAda ke*
- *bhAratiya ballebAja* CRIC *ne 100 rana*
- *mere vichAra se* CRIC *ko Takkara dene*

4.3 Pattern Quality Measure

We measured the quality of a pattern depending on its precision and coverage. Precision is the ratio of correct identification and the total identification. If the precision is high then also we have assumed that the pattern is a $good$ pattern. In our development we have marked a pattern as $good$ if the precision is 100%.

We search the initial patterns in the corresponding 'part' corpus to measure the precision and coverage. If the precision is less than 100% for a pattern then we have rejected the pattern. Otherwise we have tried to make it more generalized to increase the coverage. To make the generalization we have dropped the left most and right most tokens one by one and checked the pattern quality. If for a initial pattern, several patterns presents with 100% precision then we have selected those patterns for which no subset of those is a $good$ pattern. By this way we have prepared a list of $good$ patterns for a particular gazetteer type.

In time of 'good' pattern selection we have made some interesting observations.

- There are some patterns which satisfy the 100% precision criteria but the coverage is very poor in terms of new entity extraction. For example, "mAsTara blAsTara $CRIC$ ko" is a pattern with 100% precision. The pattern has 24 instances in the 'part' corpus, but all the extracted entities are 'sachina tedulkara'. We have also examined the pattern in the total raw corpus. It is capable of extracting 'sachina tedulkara' only. So in spite of fulfilling all the criteria the pattern is not a 'good' pattern.
- Another interesting observation is, that there are some patterns which are 'good' patterns in

the context of the ‘part’ corpus, but when used in the total raw corpus, it extracts non-relevant entities. For example “mere vichAra se $CRIC$ ko Takkara” is a ‘good’ pattern so it should extract the names of the cricketers. But when this is used in the total raw corpus it extracts non-cricketer entities (e.g. tennis players, chess players) also. To make such patterns useful we have extracted all the cricket related sentences in the similar way which was used for selecting the ‘part’ corpus and then the patterns are used to extract entities from these sentences.

- There present are patterns with very high coverage but precision is just below 100%. We have analyzed these patterns and manually identified the wrongly extracted entities. If the wrong entities can be grouped together and are having some specific properties then we have added these entities in a ‘pattern exception list’. Then the pattern is used as a ‘good’ pattern and the exception list is used to detect the wrong identifications.

In the following we have given some example of ‘good’ patterns which are useful in identification of the names of the cricket players.

- ballebAja $CRIC$ ko Tima ke
- ballebAja $CRIC$ ne
- $CRIC$ kA arddhashataka

4.4 Gazetteer Preparation

The extracted ‘good’ patterns are capable of identifying NEs from a raw corpus. These patterns are then used to prepare the gazetteers. The seeds form the initial gazetteer list for a particular gazetteer type. The ‘good’ patterns are used to extract entities from the total raw corpus. The entities identified by the patterns are added to the corresponding gazetteer list. In that way we can add more entities in our first phase gazetteer list. These new entities are taken as seeds for the next phase. Then the same procedure is followed repeatedly to develop a large gazetteer.

We have already mentioned that we have worked with a sports domain corpus and prepared some gazetteers. This gazetteers are prepared just to prove the efficiency of our approach. By using only 3 seed entities we become able to prepare a gazetteer which contains 412 names of the cricketers. Even using this approach only one seed ‘Sachin Tendualkar’ extracts 297 names after the second iteration. Similarly we have collected 245 names of tennis players from 5 seed entities.

5 Conclusion

In this paper we have described our approaches for the preparation of gazetteers. We have also prepared some gazetteers using both the approaches to show their effectiveness. These approaches are very useful for the NER task in resource-poor languages and also in domain specific NER task.

References

Al-Onaizan Y. and Knight K. 2002. Machine Transliteration of Names in Arabic Text. *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*.

Bikel Daniel M., Miller Scott, Schwartz Richard and Weischedel Ralph. 1997. Nymble: A high performance learning name-finder. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, peges 194–201.

Borthwick Andrew. 1999. A Maximum Entropy Approach to Named Entity Recognition. *Ph.D. thesis, Computer Science Department, New York University*.

Downey D., Etzioni O., Soderland S., Weld D.S. 2004. Learning text patterns for Web information extraction and assessment. In *AAAI-04 Workshop on Adaptive Text Extraction and Mining*, pages 50–55.

Ekbal A. and Bandyopadhyay S. 2007. Lexical Pattern Learning from Corpus Data for Named Entity Recognition. In *Proceedings of International Conference on Natural Language Processing (ICON), 2007.*

Ekbal A., Naskar S. and Bandyopadhyay S. 2006. A Modified Joint Source Channel Model for Transliteration. In *Proceedings of the COLING/ACL 2006, Australia*, pages 191–198.

Etzioni Oren, Cafarella Michael, Downey Doug, Popescu Ana-Maria, Shaked Tal, Soderland Stephen, Weld Daniel S. and Yates Alexander. 2005. Unsupervised named-entity extraction from the Web: An experimental study. In *Artificial Intelligence*, 165(1): 91-134.

Grishman Ralph. 1995. The New York University System MUC-6 or Where’s the syntax? In *Proceedings of the Sixth Message Understanding Conference*.

Knight K. and Graehl J. 1998. Machine Transliteration. *Computational Linguistics*, 24(4): 599–612.

Kozareva Zornitsa. 2006. Bootstrapping Named Entity Recognition with Automatically Generated Gazetteer Lists. In *Proceedings of EACL student session (EACL 2006)*.

Li Wei and McCallum Andrew. 2004. Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction (Short Paper). In *ACM Transactions on Computational Logic*.

Lin Winston, Yangarber Roman and Grishman Ralph. 2003. Bootstrapped learning of semantic classes from positive and negative examples. In *Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*.

Riloff E. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Articial Intelligence*, pages 1044–1049.

Srihari R., Niu C. and Li W. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. In *Proceedings of the sixth conference on Applied natural language processing*.

Soderland Stephen, Fisher David, Aseltine Jonathan, Lehnert Wendy. 1995. CRYSTAL: Inducing a Conceptual Dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*.

Talukdar P. Pratim, T. Brants, M. Liberman and F. Pereira. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*.

Wakao T., Gaizauskas R. and Wilks Y. 1996. Evaluation of an algorithm for the recognition and classification of proper names. In *Proceedings of COLING-96*.