Named Entity Recognition Using Appropriate Unlabeled Data, Post-processing and Voting

Asif Ekbal and Sivaji Bandyopadhyay
Department of Computer Science and Engineering
Jadavpur University
Kolkata, 700032, India
E-mail: asif.ekbal@gmail.com and sivaji_cse_ju@yahoo.com

Keywords: named entity recognition, maximum entropy, conditional random ield, support vector machine, weighted voting, Bengali

Received: January 10, 2009

This paper reports how the appropriate unlabeled data, post-processing and voting can be effective to improve the performance of a Named Entity Recognition (NER) system. The proposed method is based on a combination of the following classifiers: Maximum Entropy (ME), Conditional Random Field (CRF) and Support Vector Machine (SVM). The training set consists of approximately 272K wordforms. The proposed method is tested with Bengali. A semi-supervised learning technique has been developed that uses the unlabeled data during training of the system. We have shown that simply relying upon the use of large corpora during training for performance improvement is not in itself sufficient. We describe the measures to automatically select effective documents and sentences from the unlabeled data. In addition, we have used a number of techniques to post-process the output of each of the models in order to improve the performance. Finally, we have applied weighted voting approach to combine the models. Experimental results show the effectiveness of the proposed approach with the overall average recall, precision, and f-score values of 93.79%, 91.34%, and 92.55%, respectively, which shows an improvement of 19.4% in f-score over the least performing baseline ME based system and an improvement of 15.19% in f-score over the best performing baseline SVM based system.

Povzetek: Razvita je metoda za prepoznavanje imen, ki temelji na uteženem glasovanju več klasifikatorjev.

1 Introduction

Named Entity Recognition (NER) is an important tool in almost all Natural Language Processing (NLP) application areas such as Information Extraction [1], Machine Translation [2], Question Answering [3] etc. The objective of NER is to identify and classify every word/term in a document into some predefined categories like person name, location name, organization name, miscellaneous name (date, time, percentage and monetary expressions etc.) and "none-of-the-above". The challenge in detection of named entities (NEs) is that such expressions are hard to analyze using rule-based NLP because they belong to the open class of expressions, i.e., there is an infinite variety and new expressions are constantly being invented.

In recent years, automatic NER systems have become a popular research area in which a considerable number of studies have been addressed on developing these systems. These can be classified into three main classes [4], namely rule-based NER, machine learning-based NER and hybrid NER.

Rule-based approaches focus on extracting names using a number of hand-crafted rules. Generally, these systems consist of a set of patterns using grammatical (e.g., part of speech), syntactic (e.g., word precedence) and orthographic features (e.g., capitalization) in combination with dictionaries [5]. A NER system has been proposed in [6][7] based on carefully handcrafted regular expression called FASTUS. They divided the task into three steps: recognizing phrase, recognizing patterns and merging incidents, while [8] uses extensive specialized resources such as gazetteers, white and yellow pages. The NYU system [9] was introduced that uses handcrafted rules. A rule-based Greek NER system [10] has been developed in the context of the R&D project MITOS [1]. The NER system consists of three processing stages: linguistic pre-processing, NE identification and NE classification. The linguistic pre-processing stage involves some basic tasks: tokenisation, sentence splitting, part of speech (POS) tagging and stemming. Once the text has been annotated with POS tags, a stemmer is used. The aim of the stemmer is to reduce the size of the lexicon as well as the size and complexity of NER grammar. The NE identification phase involves the detection of their boundaries, i.e., the start and end of all the possible spans of tokens that are likely to belong to a NE. Classification involves three sub-stages: application of classification rules, gazetteer-based classification, and par-

[1] http://www.iit.demokritos.gr/skel/mitos

tial matching of classified NEs with unclassified ones. The French NER system has been implemented with the rule-based inference engine [11]. It is based on a large knowledge base including 8,000 proper names that share 10,000 forms and consists of 11,000 words. It has been used continuously since 1995 in several real-time document filtering applications [12]. Other rule-based NER systems are University Of Sheffield's LaSIE-II [13], ISOQuest's NetOwl [14] and University Of Edinburgh's LTG [15] [16] for English NER. These approaches are relying on manually coded rules and compiled corpora. These kinds of systems have better results for restricted domains and are capable of detecting complex entities that are difficult with learning models. However, rule-based systems lack the ability of portability and robustness, and furthermore the high cost of the maintenance of rules increases even though the data is slightly changed. These types of systems are often domain dependent, language specific and do not necessarily adapt well to new domains and languages.

Nowadays, machine-learning (ML) approaches are popularly used in NER because these are easily trainable, adaptable to different domains and languages as well as their maintenance are also less expensive [17]. On the other hand, rule-based approaches lack the ability of coping with the problems of robustness and portability. Each new source of text requires significant tweaking of rules to maintain optimal performance and the maintenance costs could be quite high. Some of the well-known machine-learning approaches used in NER are Hidden Markov Model (HMM)(BBN's IdentiFinder [18] [19]), Maximum Entropy (ME)(New York University's MENE in ([20]; [21]), Decision Tree (New York University's system in [22] and SRA's system in [23] and CRF [24]; [25]. Support Vector Machines (SVMs) based NER system was proposed by Yamada et al. [26] for Japanese. His system is an extension of Kudo's chunking system [27] that gave the best performance at CoNLL-2000 shared tasks. The other SVM-based NER systems can be found in [28] and [29].

Unsupervised learning method is another type of machine learning model, where an unsupervised model learns without any feedback. In unsupervised learning, the goal is to build representations from data. [30] discusses an unsupervised model for NE classification by the use of unlabeled examples of data. An unsupervised NE classification models and their ensembles have been introduced in [31] that uses a small-scale NE dictionary and an unlabeled corpus for classifying NEs. Unlike rule-based models, these types of models can be easily ported to different domains or languages.

In hybrid systems, the goal is to combine rule-based and machine learning-based methods, and develop new methods using strongest points from each method. [32] described a hybrid document centered system, called LTG system. [33] introduced a hybrid system by combining HMM, MaxEnt and handcrafted grammatical rules. Although, this approach can get better result than some other approaches, but weakness of handcraft rule-based NER surfaces when there is a need to change the domain of data. Previous works [34, 35] have also shown that combining several ML models using voting technique always performs better than any single ML model.

When applying machine-learning techniques to NLP tasks, it is time-consuming and expensive to hand-label the large amounts of training data necessary for good performance. In the literature, we can find the use of unlabeled data in improving the performance of many tasks such as name tagging [36], semantic class extraction [37] and co-reference resolution [38]. However, it is important to decide how the system should effectively select unlabeled data, and how the size and relevance of data impact the performance. A technique to automatically select documents is reported in [39].

India is a multilingual country with great cultural diversities. However, the relevant works in NER involving Indian languages have started to appear very recently. Named Entity (NE) identification in Indian languages in general and Bengali in particular is difficult and challenging as:

1. Unlike English and most of the European languages, Bengali lacks capitalization information, which plays a very important role in identifying NEs.
2. Indian person names are more diverse compared to the other languages and a lot of these words can be found in the dictionary with some other specific meanings.
3. Bengali is a highly inflectional language providing one of the richest and most challenging sets of linguistic and statistical features resulting in long and complex wordforms.
4. Bengali is a relatively free order language.
5. Bengali, like other Indian languages, is a resource poor language - annotated corpora, name dictionaries, good morphological analyzers, POS taggers etc. are not yet available in the required measure.
6. Although Indian languages have a very old and rich literary history, technological developments are of recent origin.
7. Web sources for name lists are available in English, but such lists are not available in Bengali forcing the use of transliteration for creating such lists.

A pattern directed shallow parsing approach for NER in Bengali has been reported in [40]. The paper reports about two different NER models, one using the lexical contextual patterns and the other using the linguistic features along with the same set of lexical contextual patterns. A HMM-based NER system has been reported in [41], where more contextual information has been considered during the emission probabilities and NE suffixes have been kept for handling the unknown words. More recently, the works in the area of Bengali NER can be found in [42] with ME, in [43] with CRF and in [44] with SVM approach. These

systems were developed with the help of a number of features and gazetteers. The method of improving the performance of NER system using appropriate unlabeled data, post-processing and voting has been reported in [45].

Other than Bengali, the works on Hindi can be found in [46] with CRF model using feature induction technique to automatically construct the features that does a maximal increase in the conditional likelihood. A language independent method for Hindi NER has been reported in [47]. Sujan et al. [48] reported a ME based system with the hybrid feature set that includes statistical as well as linguistic features. A MEMM-based system has been reported in [49]. As part of the IJCNLP-08 NER shared task, various works of NER in Indian languages using various approaches can be found in IJCNLP-08 NER Shared Task on South and South East Asian Languages (NERSSEAL)[2]. As part of this shared task, [50] reported a CRF-based system followed by post-processing which involves using some heuristics or rules. A CRF-based system has been reported in [51], where it has been shown that the hybrid HMM model can perform better than CRF.

Srikanth and Murthy [52] developed a NER system for Telugu and tested it on several data sets from the Eenaadu and Andhra Prabha newspaper corpora. They obtained the overall f-measure between 80-97% with person, location and organization tags. For Tamil, a CRF-based NER system has been presented in [53] for the tourism domain. This approach can take care of morphological inflections of NEs and can handle nested tagging with a hierarchical tagset containing 106 tags. Shishtla et al. [54] developed a CRF-based system for English, Telugu and Hindi. They suggested that character n-gram based approach is more effective than the word based models. They described the features used and the experiments to increase the recall of NER system.

In this paper, we have reported a NER system for Bengali by combining the outputs of the classifiers, namely ME, CRF and SVM. In terms of native speakers, Bengali is the seventh most spoken language in the world, second in India and the national language of Bangladesh. We have manually annotated a portion of the Bengali news corpus, developed from the web-archive of a leading Bengali newspaper with *Person name*, *Location name*, *Organization name* and *Miscellaneous name* tags. We have also used the IJCNLP-08 NER Shared Task data that was originally annotated with a fine-grained NE tagset of twelve tags. This data has been converted into the forms to be tagged with NEP (Person name), NEL (Location name), NEO (Organization name), NEN (Number expressions), NETI (Time expressions) and NEM (Measurement expressions). The NEN, NETI and NEM tags are mapped to point to the miscellaneous entities. The system makes use of the different contextual information of the words along with the variety of orthographic word level features that are helpful in predicting the various NE classes. We have considered both language independent as well as language dependent features. Language independent features are applicable to almost all the languages including Bengali and Hindi. Language dependent features have been extracted from the language specific resources such as the part of speech (POS) taggers and gazetteers. It has been observed from the evaluation results that the use of language specific features improves the performance of the system. We also conducted a number of experiments to find out the best-suited set of features for NER in each of the languages. We have developed an unsupervised method to generate the lexical context patterns that are used as the features of the classifiers. A semi-supervised technique has been proposed to select the appropriate unlabeled documents from a large collection of unlabeled corpus. The main contribution of this work is as follows:

1. An unsupervised technique has been reported to generate the context patterns from the unlabeled corpus.
2. A semi-supervised ML technique has been developed in order to use the unlabeled data.
3. Relevant unlabeled documents are selected using CRF techniques. We have selected effective sentences to be added to the initial labeled data by applying majority voting between ME model, CRF and two different models of SVM. In the previous literature [39], the use of any single classifier was reported for selecting appropriate sentences.
4. Useful features for NER in Bengali are identified. A number of features are language independent and can be applicable to other languages also.
5. The system has been evaluated in two different ways: Without language dependent features and with language dependent features.
6. Three different post-processing techniques have been reported in order to improve the performance of the classifiers.
7. Finally, models are combined using three weighted voting techniques.

2 Named entity tagged corpus development

The rapid development of language resources and tools using machine learning techniques for less computerized languages requires appropriately tagged corpus. There is a long history of creating a standard for western language resources. The human language technology (HLT) society in Europe has been particularly zealous for the standardization of European languages. On the other hand, in spite of having great linguistic and cultural diversity, Asian language resources have received much less attention than their western counterparts. India is a multilingual country with a diverse cultural heritage. Bengali is one of the most

[2] http://ltrc.iiit.ac.in/ner-ssea-08

popular languages and predominantly spoken in the eastern part of India. In terms of native speakers, Bengali is the seventh most spoken language in the World, second in India and the national language in Bangladesh. In the literature, there has been no initiative of corpus development from the web in Indian languages and specifically in Bengali.

Newspaper is a huge source of readily available documents. Web is a great source of language data. In Bengali, there are some newspapers (like, Anandabazar Patrika, Bartaman, Dainik, Ittefaq etc.), published from Kolkata and Bangladesh, which have their internet-edition in the web and some of them provide their archive available also. A collection of documents from the archive of the newspaper, stored in the web, may be used as the corpus, which in turn can be used in many NLP applications.

We have followed the method of developing the Bengali news corpus in terms of language resource acquisition using a web crawler, language resource creation that includes HTML file cleaning, code conversion and language resource annotation that involves defining a tagset and subsequent tagging of the news corpus. A web crawler has been designed that retrieves the web pages in Hyper Text Markup Language (HTML) format from the news archive. Various types of news (International, National, State, Sports, Business etc.) are collected in the corpus and so a variety of linguistics features of Bengali are covered. The Bengali news corpus is available in UTF-8 and contains approximately 34 million wordforms.

A news corpus, whether in Bengali or in any other language has different parts like title, date, reporter, location, body etc. To identify these parts in a news corpus the tagset described in Table 1 have been defined. Detailed of this corpus development work can be found in [55].

The *date*, *location*, *reporter* and *agency* tags present in the web pages of the Bengali news corpus have been automatically named entity (NE) tagged. These tags can identify the NEs that appear in some fixed places of the newspaper. In order to achieve reasonable performance for NER, supervised machine learning approaches are more appropriate and this requires a completely tagged corpus. This requires the selection of an appropriate NE tagset.

With respect to the tagset, the main feature that concerns us is its granularity, which is directly related to the size of the tagset. If the tagset is too coarse, the tagging accuracy will be much higher, since only the important distinctions are considered, and the classification may be easier both by human manual annotators as well as the machine. But, some important information may be missed out due to the coarse grained tagset. On the other hand, a too fine-grained tagset may enrich the supplied information but the performance of the automatic named entity tagger may decrease. A much richer model is required to be designed to capture the encoded information when using a fine grained tagset and hence, it is more difficult to learn.

When we are about to design a tagset for the NE disambiguation task, the issues that need consideration include - the type of applications (some application may required more complex information whereas only category information may be sufficient for some tasks), tagging techniques to be used (statistical, rule based which can adopt large tagsets very well, supervised/unsupervised learning). Further, a large amount of annotated corpus is usually required for statistical named entity taggers. A too fine grained tagset might be difficult to use by human annotators during the development of a large annotated corpus. Hence, the availability of resources needs to be considered during the design of a tagset.

Table 2: Statistics of the NE tagged corpus

Total Number of sentences	23,181
Number of wordforms (approx.)	200K
Number of NEs	19,749
Average length of NE	2 (approx.)

During the design of the tagset for Bengali, our main aim was to build a small but clean and completely tagged corpora for Bengali. The resources can be used for the conventional usages like Information Retrieval, Information Extraction, Event Tracking System, Web People Search etc. We have used CoNLL 2003 shared task tagset as reference point for our tagset design.

We have used a NE tagset that consists of the following four tags:

1. *Person name*: Denotes the names of people. For example, *sachin*[Sachin] /*Person name*, *manmohan singh*[Manmohan Singh]/*Person name*.

2. *Location name*: Denotes the names of places. For example, *jadavpur*[Jadavpur]/*Location name*, *new delhi*[New Delhi]/*Location name*.

3. *Organization name*: Denotes the names of organizations. For example, *infosys*[Infosys]/*Organization name*, *jadavpur vishwavidyalaya*[Jadavpur University]/*Organization name*.

4. *Miscellaneous name*: Denotes the miscellaneous NEs that include date, time, number, monetary expressions, measurement expressions and percentages. For example, *15th august 1947*[15th August 1947]/*Miscellaneous name*, *11 am*[11 am]/*Miscellaneous name*, 110/*Miscellaneous name*, *1000 taka*[1000 rupees]/*Miscellaneous name*, *100%*[100%]/ *Miscellaneous name* and *100 gram*[100 gram]/ *Miscellaneous name*.

We have manually annotated approximately 200K wordforms of the Bengali news corpus.The annotation has been carried out by one expert and edited by another expert. The corpus is in the Shakti Standard Format (SSF) form [56]. Some statistics of this corpus is shown in Table 2.

We have also used the NE tagged corpus of the IJCNLP Shared Task on Named Entity Recognition for South

Table 1: News corpus tag set

Tag	Definition	Tag	Definition
header	Header of the news document	reporter	Reporter-name
title	Headline of the news document	agency	Agency providing news
t1	1st headline of the title	location	The news location
t2	2nd headline of the title	body	Body of the news document
date	Date of the news document	p	Paragraph
bd	Bengali date	table	Information in tabular form
day	Day	tc	Table Column
ed	English date	tr	Table row

Table 3: Statistics of the IJCNLP-08 NE tagged corpus

Total Number of sentences	7035
Number of wordforms (approx.)	122K
Number of NEs	5921
Average length of NE	2 (approx.)

and South East Asian Languages (NERSSEAL)[3]. A fine grained tagset of twelve tags were defined as part of this shared task. The underlying reason to adopt this finer NE tagset is to use the NER system in various NLP applications, particularly in machine translation. The IJCNLP-08 NER shared task tagset is shown in Table 4. One important aspect of the shared task was to identify and classify the maximal NEs as well as the nested NEs, i.e, the constituent part of a larger NE. But, the training data were provided with the type of the maximal NE only. For example, *mahatma gandhi road* (Mahatma Gandhi Road) was annotated as location and assigned the tag 'NEL' even if *mahatma* (Mahatma) and *gandhi*(Gandhi) are NE title person (NETP), and person name (NEP), respectively. The task was to identify *mahatma gandhi road* as a NE and classify it as NEL. In addition, *mahatma*, and *gandhi* were to be recognized as NEs of the categories NETP (Title person) and NEP (Person name) respectively. Some NE tags are hard to distinguish in some contexts. For example, it is not always clear whether something should be marked as 'Number' or as 'Measure'. Similarly, 'Time' and 'Measure' is another confusing pair of NE tags. Another difficult class is 'Technical terms' and it is often confusing whether any expression is to be tagged as the 'NETE' (NE term expression) or not. For example, it is difficult to decide whether 'Agriculture' is 'NETE', and if not then whether 'Horticulture' is 'NETE' or not. In fact, this the most difficult class to identify. Other ambiguous tags are 'NETE' and 'NETO' (NE title-objects). The corpus is in the Shakti Standard Format (SSF) form [56]. We have also manually annotated a portion of the Bengali news corpus [55] with the twelve NE tags of the shared task tagset. Some statistics of this corpus is shown in Table 3.

We have considered only those NE tags that denote person name, location name, organization name, number expression, time expression and measurement expressions. The number, time and measurement expressions are mapped to belong to the *Miscellaneous name* tag. Other tags of the shared task have been mapped to the 'other-than-NE' category. Hence, the final tagset is shown in Table 5.

In order to properly denote the boundaries of the NEs, the four NE tags are further subdivided as shown in Table 6. In the output, these sixteen NE tags are directly mapped to the four major NE tags, namely *Person name*, *Location name*, *Organization name* and *Miscellaneous name*.

3 Named entity recognition in Bengali

In terms of native speakers, Bengali is the seventh popular language in the world, second in India and the national language of Bangladesh. We have used a Bengali news corpus [55], developed from the web-archive of a widely read Bengali newspaper for NER. A portion of this corpus containing 200K wordforms has been manually annotated with the four NE tags namely, *Person name*, *Location name*, *Organization name* and *Miscellaneous name*. The data has been collected from the International, National, State and Sports domains. We have also used the annotated corpus of 122K wordforms, collected from the IJCNLP-08 NERSSEAL (http://ltrc.iiit.ac.in/ner-ssea-08). This data was a mixed one and dealt mainly with the literature, agriculture and scientific domains. Moreover, this data was originally annotated with a fine-grained NE tagset of twelve tags. An appropriate tag conversion routine has been defined as shown in Table 5 in order to convert this data into the desired forms, tagged with the four NE tags.

3.1 Approaches

NLP research around the world has taken giant leaps in the last decade with the advent of effective machine learning algorithms and the creation of large annotated corpora for various languages. However, annotated corpora and other lexical resources have started appearing only very recently in India. In this paper, we have reported a NER system by combining the outputs of the classifiers, namely ME, CRF

[3]http://ltrc.iiit.ac.in/ner-ssea-08

Table 4: Named entity tagset for Indian languages (IJCNLP-08 NER Shared Task Tagset)

NE Tag	Meaning	Example
NEP	Person name	*sachin*/NEP, *sachin ramesh tendulkar* / NEP
NEL	Location name	*kolkata*/NEL, *mahatma gandhi road*/ NEL
NEO	Organization name	*jadavpur bishbidyalya*/NEO, *bhaba eytomik risarch sentar* / NEO
NED	Designation	*chairrman*/NED, *sangsad*/NED
NEA	Abbreviation	*b a*/NEA, *c m d a*/NEA, *b j p*/NEA, *i.b.m*/ NEA
NEB	Brand	*fanta*/NEB
NETP	Title-person	*shriman*/NED, *shri*/NED, *shrimati*/NED
NETO	Title-object	*american beauty*/NETO
NEN	Number	*10*/NEN, *dash*/NEN
NEM	Measure	*tin din*/NEM, *panch keji*/NEM
NETE	Terms	*hiden markov model*/NETE, *chemical reaction*/NETE
NETI	Time	*10 i magh 1402* / NETI, *10 am*/NETI

Table 5: Tagset used in this work

IJCNLP-08 shared task tagset	Tagset used	Meaning
NEP	*Person name*	Single word/multiword person name
NEL	*Location name*	Single word/multiword location name
NEO	*Organization name*	Single word/multiword organization name
NEN, NEM, NETI	*Miscellaneous name*	Single word/ multiword miscellaneous name
NED, NEA, NEB, NETP, NETE	NNE	Other than NEs

Table 6: Named entity tagset (B-I-E format)

Named Entity Tag	Meaning	Example
PER	Single word person name	*sachin*/PER, *rabindranath*/PER
LOC	Single word location name	*kolkata*/LOC, *mumbai*/LOC
ORG	Single word organization name	*infosys*/ORG
MISC	Single word miscellaneous name	*10*/MISC, *dash*/MISC
B-PER I-PER E-PER	Beginning, Internal or the End of a multiword person name	*sachin*/B-PER *ramesh*/I-PER *tendulkar* /E-PER, *rabindranath*/B-PER *thakur*/E-PER
B-LOC I-LOC E-LOC	Beginning, Internal or the End of a multiword location name	*mahatma*/B-LOC *gandhi* /I-LOC *road* /E-LOC, *new*/B-LOC *york*/E-LOC
B-ORG I-ORG E-ORG	Beginning, Internal or the End of a multiword organization name	*jadavpur* /B-ORG *bishvidyalya*/E-ORG, *bhaba* /B-ORG *eytomik*/I-ORG *risarch*/I-ORG *sentar* /E-ORG
B-MISC I-MISC E-MISC	Beginning, Internal or the End of a multiword miscellaneous name	*10 i* /B-MISC *magh*/I-MISC *1402*/E-MISC, *10*/B-MISC *am*/E-MISC
NNE	Other than NEs	*kara*/NNE, *jal*/NNE

and SVM frameworks in order to identify NEs from a Bengali text and to classify them into *Person name*, *Location name*, *Organization name* and *Miscellaneous name*. We have developed two different systems with the SVM model, one using **forward parsing** (SVM-F) that parses from left to right and other using **backward parsing** (SVM-B) that parses from right to left. The SVM system has been developed based on [57], which perform classification by constructing a N-dimensional hyperplane that optimally separates data into two categories. We have used *YamCha* toolkit (http://chasen-org/∼taku/software/yamcha), an SVM based tool for detecting classes in documents and formulating the NER task as a sequence labeling problem. Here, the pair wise multi-class decision method and *polynomial kernel function* have been used. We have used TinySVM-0.0[4] TinySVM classifier that seems to be the best optimized among publicly available SVM toolkits. We have used the Maximum Entropy package (http://homepages.inf.ed.ac.uk/s0450736/software/maxent/maxent-20061005.tar.bz2). We have used C++ based CRF++ package (http://crfpp.sourceforge.net) for NER.

During testing, it is possible that the classifier produces a sequence of inadmissible classes (e.g., B-PER followed by LOC). To eliminate such sequences, we define a transition probability between word classes $P(c_i|c_j)$ to be equal to 1 if the sequence is admissible, and 0 otherwise. The probability of the classes $c_1, c_2, \ldots, c_n$ assigned to the words in a sentence 's' in a document 'D' is defined as follows: $P(c_1, c_2, \ldots, c_n|S, D) = \prod_{i=1}^{n} P(c_1|S, D) \times P(c_i|c_{i-1})$ where $P(c_1|S, D)$ is determined by the ME/CRF/SVM classifier.

Performance of the NER models has been limited in part by the amount of labeled training data available. We have used unlabeled corpus to address this problem. Based on the original training on the labeled corpus, there will be some tags in the unlabeled corpus that the taggers will be very sure about. For example, there will be contexts that were always followed by a person name (*sri*, *mr.* etc.) in the training corpus. While a new word W is found in this context in the unlabeled corpus then it can be predicted as a person name. If any tagger can learn this fact about W, it can successfully tag W when it appears in the test corpus without any indicative context. In the similar way, if a previously unseen context appears consistently in the unlabeled corpus before known NE then the tagger should learn that this is a predicative context. We have developed a semi-supervised learning approach in order to capture this information that are used as the features in the classifiers. We have used another semi-supervised learning approach in order to select appropriate data from the available large unlabeled corpora and added to the initial training set in order to improve the performance of the taggers. The models are retrained with this new training set and this process is repeated in a bootstrapped manner.

[4] http://cl.aist-nara.ac.jp/∼taku ku/software/

We have also used a number of post-processing rules in order to improve the performance in each of the models. Finally, three models are combined together into a single system with the help of three weighted voting schemes.

In the following subsections, some of our earlier attempts in NER have been reported that form the base of our overall approach in NER.

3.1.1 Pattern directed shallow parsing approach

Two NER models, namely A and B, using a pattern directed shallow parsing approach have been reported in [40]. An unsupervised algorithm has been developed that tags the unlabeled corpus with the seed entities of *Person name*, *Location name* and *Organization name*. These seeds have been prepared by automatically extracting the words from the *reporter*, *location* and *agency* tags of the Bengali news corpus [55]. Model A uses only the seed lists to tag the training corpus whereas in model B, we have used the various gazetteers along with the seed entities for tagging. The lexical context patterns generated in such way are used to generate further patterns in a bootstrapped manner. The algorithm terminates until no new patterns can be generated. During testing, model A can not deal with the *NE classification disambiguation* problem (i.e, can not handle the situation when a particular word is tagged with more than one NE type) but model B can handle with this problem with the help of gazetteers and various language dependent features.

3.1.2 HMM based NER system

A HMM-based NER system has been reported in [41], where more context information has been considered during emission probabilities and the word suffixes have been used for handling the unknown words. A brief description of the system is given below:

In the HMM based NE tagging, the task is to find the sequence of NE tags $T = t_1, t_2, t_3, \ldots t_n$ that is optimal for a word sequence $W = w_1, w_2, w_3 \ldots w_n$. The tagging problem becomes equivalent to searching for $argmax_T P(T) * P(W|T)$, by the application of Bayes' law.

A trigram model has been used for transition probability, that is, the probability of a tag depends on two previous tags, and then we have,

$P(T) = P(t_1|\$) \times P(t_2|\$, t_1) \times P(t_3|t_1, t_2) \times P(t_4|t_2, t_3) \times \ldots \times P(t_n|t_{n-2}, t_{n-1})$

where an additional tag '$' (dummy tag) has been introduced to represent the beginning of a sentence. Due to sparse data problem, the linear interpolation method has been used to smooth the trigram probabilities as follows:

$P'(t_n|t_{n-2}, t_{n-1}) = \lambda_1 P(t_n) + \lambda_2 P(t_n|t_{n-1}) + \lambda_3 P(t_n|t_{n-2}, t_{n-1})$

such that the λs sum to 1. The values of λs have been calculated by the method given in [58].

Additional context dependent feature has been introduced to the emission probability to make the Markov model more powerful. The probability of the current word depends on the tag of the previous word and the tag to be assigned to the current word. Now, we calculate $P(W|T)$ by the following equation:

$P(W|T) \approx P(w_1|\$, t_1) \times P(w_2|t_1, t_2) \times \ldots \times P(w_n|t_{n-1}, t_n)$.

So, the emission probability can be calculated as:

$P(w_i|t_{i-1}, t_i) = \frac{freq(t_{i-1}, t_i, w_i)}{freq(t_{i-1}, t_i)}$

Here, also the smoothing technique is applied rather than using the emission probability directly. The emission probability is calculated as:

$P'(w_i|t_{i-1}, t_i) = \theta_1 P(w_i|t_i) + \theta_2 P(w_i|t_{i-1}, t_i)$,

where θ_1, θ_2 are two constants such that all θs sum to 1. In general, the values of θs can be calculated by the same method that was adopted in calculating λs.

Handling of unknown words is an important problem in the HMM based NER system. For words which have not been seen in the training set, $P(w_i|t_i)$ is estimated based on features of the unknown words, such as whether the word contains a particular suffix. The list of suffixes has been prepared that usually appear at the end of NEs. A null suffix is also kept to take care of those words that have none of the suffixes in the list. The probability distribution of a particular suffix with respect to specific NE tag is generated from all words in the training set that share the same suffix.

Incorporating diverse features in an HMM based NE tagger is difficult and complicates the smoothing typically used in such taggers. Indian languages are morphologically very rich and contains a lot of non-independent features. A ME [20] or CRF [25] or SVM [26] based method can deal with the diverse and overlapping features of the Indian languages more efficiently than HMM.

3.1.3 Other NER sytems

A ME based NER system for Bengali has been reported in [42]. The system has been developed with the contextual information of the words along with the variety of orthographic word-level features. In addition, a number of manually developed gazetteers have been used as the features in the model. We conducted a number of experiments in order to find out the appropriate features for NER in Bengali. Detailed evaluation results have shown the best performance with a contextual word window of size three, i.e., previous word, current word and the next one word, dynamic NE tag of the previous word, POS tag of the current word, prefixes and suffixes of length up to three characters of the current word and binary valued features extracted from the gazetteers.

A CRF based NER system has been described in [43]. The system has been developed with the same set of features as that of ME. Evaluation results have demonstrated the best results with a contextual window of size five, i.e, previous two words, current word and next two words, NE tag of the previous word, POS tags of the current and the previous words, suffixes and prefixes of length up to three characters of the current word, and the various binary valued features extracted from the several gazetteers.

A SVM based NER system has been described in [44]. This model also makes use of the different contextual information of the words, orthographic word-level features along with the various gazetteers. Results have demonstrated the best results with a contextual window of size six, i.e., previous three words, current word and next two words, NE tag of the previous two words, POS tags of the current, previous word and the next words, suffixes and prefixes of length of length up to three characters of the current word, and the various binary valued features extracted from the several gazetteers.

4 Named entity features

Feature selection plays a crucial role in any statistical model. ME model does not provide a method for automatic selection of given feature sets. Usually, heuristics are used for selecting effective features and their combinations. It is not possible to add arbitrary features in a ME framework as that will result in overfitting. Unlike ME, CRF does not require careful feature selection in order to avoid overfitting. CRF has the freedom to include arbitrary features, and the ability of feature induction to automatically construct the most useful feature combinations. Since, CRFs are log-linear models, and high accuracy may require complex decision boundaries that are non-linear in the space of original features, the expressive power of the models is often increased by adding new features that are conjunctions of the original features. For example, a conjunction feature might ask if the current word is in the person name list and the next word is an action verb '*ballen*'(told). One could create arbitrary complicated features with these conjunctions. However, it is infeasible to incorporate all possible conjunctions as these might result in overflow of memory as well as overfitting. Support vector machines predict the classes depending upon the labeled word examples only. It predicts the NEs based on feature information of words collected in a predefined window size while ME or CRF predicts them based on the information of the whole sentence. So, CRF can handle the NEs with outside tokens, which SVM always tags as 'NNE'. A CRF has different characteristics from SVM, and is good at handling different kinds of data. In particular, SVMs achieve high generalization even with training data of a very high dimension. Moreover, with the use of *kernel function*, SVMs can handle non-linear feature spaces, and carry out the training considering combinations of more than one feature.

The main features for the NER task have been identified based on the different possible combination of available word and tag context. The features also include prefix and suffix for all words. The term prefix/suffix is a sequence of first/last few characters of a word, which may not be a linguistically meaningful prefix/suffix. The use of prefix/suffix information works well for the highly inflected languages as like the Indian languages. In addition to these, various gazetteer lists have been developed for use in the NER tasks. We have considered different combination from the following set for inspecting the best set of features for NER in Bengali:

F={$w_{i-m}, \ldots, w_{i-1}, w_i, w_{i+1}, \ldots w_{i+n}$, |prefix| $\leq n$, |suffix| $\leq n$, NE tag(s) of previous word(s), POS tag(s) of the current and/or the surrounding word(s), First word, Length of the word, Digit information, Infrequent word, Gazetteer lists}, where w_i is the current word; w_{i-m} is the previous *m*th word and w_{i+n} is the next *n*th word.

The set 'F' contains both language independent as well as language dependent features. The set of language independent features includes the context words, prefixes and suffixes of all the words, NE information of the previous word(s), first word, length of the word, digit information and infrequent word. Language dependent features for Bengali include the set of known suffixes that may appear with the various NEs, clue words that help in predicting the location and organization names, words that help to recognize measurement expressions, designation words that help to identify person names, various gazetteer lists that include the first names, middle names, last names, location names, organization names, function words, weekdays and month names. As part of language dependent features for Hindi, the system uses only the lists of first names, middle names, last names, weekdays, month names along with the list of words that helps to recognize measurement expressions. We have also used the part of speech (POS) information of the current and/or the surrounding word(s) for Bengali.

Language independent NE features can be applied for NER in any language without any prior knowledge of that language. Though the lists or gazetteers are not theoretically language dependent, we call it as language dependent as these require apriori knowledge of any specific language for their preparation. Also, we include the POS information in the set of language dependent features as it depends on some language specific phenomenon such as person, number, tense, gender etc. For example, gender information has a crucial role in Hindi but it is not an issue in Bengali. In Bengali, a combination of non-finite verb followed by a finite verb can have several different morphosyntactic functions. For example, '*mere phellO*' [kill+non-finite throw+finite] can mean 'threw after killing' (here, '*mere*' is a sequential participle) or just 'killed' with a completive sense (where, '*mere*' is a polar verb and '*phellO*', the vector verb of a finite verb group). On the other hand, constructs like '*henshe ballO*' [smile+non-finite say+finite] might mean 'said while smiling' ('*henshe*' is functioning as an adverbial participle). Similarly, it is hard to distinguish between the adjectival participle and verbal nouns. The use of language specific features is helpful to improve the performance of the NER system. In the resource-constrained Indian language environment, the non-availability of language specific resources such as POS taggers, gazetteers, morphological analyzers etc. forces the development of such resources to use in NER systems. This leads to the necessity of apriori knowledge of the language.

4.1 Language independent features

We have considered different combinations from the set of language independent features for inspecting the best set of features for NER in Bengali. Following are the details of the features:

- Context word feature: Preceding and following words of a particular word can be used as the features. This is based on the observation that the surrounding words are very effective in the identification of NEs.
- Word suffix: Word suffix information is helpful to identify NEs. This is based on the observation that the NEs share some common suffixes. This feature can be used in two different ways. The first and the naïve one is, a fixed length (say, n) word suffix of the current and/or the surrounding word(s) can be treated as feature. If the length of the corresponding word is less than or equal to $n - 1$ then the feature values are not defined and denoted by ND. The feature value is also not defined (ND) if the token itself is a punctuation symbol or contains any special symbol or digit. The value of ND is set to 0. The second and the more helpful approach is to modify the feature as binary valued. Variable length suffixes of a word can be matched with predefined lists of useful suffixes for different classes of NEs. Various length suffixes belong to the category of language dependent features as they require language specific knowledge for their development.
- Word prefix: Word prefixes are also helpful and based on the observation that NEs share some common prefix strings. This feature has been defined in a similar way as that of the fixed length suffixes.
- Named Entity Information: The NE tag(s) of the previous word(s) has been used as the only dynamic feature in the experiment.
- First word: This is used to check whether the current token is the first word of the sentence or not. Though Bengali is a relatively free order language, the first word of the sentence is most likely a NE as it appears in the subject position most of the time.
- Digit features: Several binary valued digit features have been defined depending upon the presence and/or the number of digits in a token (e.g., CntDgt [token contains digits], FourDgt [four digit token], TwoDgt [two digit token]), combination of digits and punctuation symbols (e.g., CntDgtCma [token consists of digits and comma], CntDgtPrd [token consists of digits and periods]), combination of digits and symbols (e.g., CntDgtSlsh [token consists of digit and slash], CntDgtHph [token consists of digits and hyphen], CntDgtPrctg [token consists of digits and percentages]). These binary valued features are helpful in recognizing miscellaneous NEs, such as time expressions, measurement expressions and numerical numbers etc.
- Infrequent word: The frequencies of the words in the training corpus have been calculated. A cut off frequency has been chosen in order to consider the words that occur with more than the cut off frequency in the training corpus. The cut off frequency is set to 10. A binary valued feature 'Infrequent' is defined to check whether the current token appears in this list or not.
- Length of a word: This binary valued feature is used to check whether the length of the current word is less than three or not. This is based on the observation that very short words are rarely NEs.

The above set of language independent features along with their descriptions are shown in Table 7. The *baseline* models have been developed with the language independent features.

4.2 Language dependent features

Language dependent features for Bengali have been identified based on the earlier experiments [40] on NER. Additional NE features have been identified from the Bengali news corpus [55]. Various gazetteers used in the experiment are presented in Table 8. Some of the gazetteers are briefly described as below:

- NE Suffix list (variable length suffixes): Variable length suffixes of a word are matched with the predefined lists of useful suffixes that are helpful to detect person (e.g., *-babu*, *-da*, *-di* etc.) and location (e.g., *-land*, *-pur*, *-liya* etc.) names.
- Organization suffix word list: This list contains the words that are helpful to identify organization names (e.g., *kong*, *limited* etc.). These are also part of organization names.
- Person prefix word list: This is useful for detecting person names (e.g., *shriman*, *shri*, *shrimati* etc.).
- Common location word list: This list contains the words (e.g., *sarani*, *road*, *lane* etc.) that are part of the multiword location names and usually appear at their end.
- Action verb list: A set of action verbs like *balen*, *balalen*, *ballo*, *sunllO*, *hanslo* etc. often determine the presence of person names. Person names generally appear before the action verbs.
- Designation words: A list of common designation words (e.g., *neta*, *sangsad*, *kheloar* etc.) has been prepared. This helps to identify the position of person names.

Table 7: Descriptions of the language independent features. Here, *i* represents the position of the current word and w_i represents the current word

Feature	Description
ContexT	ContexT$_i = w_{i-m}, \ldots, w_{i-1}, w_i, w_{i+1}, w_{i+n}$, where w_{i-m}, and w_{i+n} are the previous m-th, and the next n-th word
Suf	Suf$_i(n) = \begin{cases} \text{Suffix string of length } n \text{ of } w_i & \text{if } \lvert w_i \rvert \geq n \\ ND(=0) & \text{if } \lvert w_i \rvert \leq (n-1) \\ & \text{or } w_i \text{ is a punctuation symbol} \\ & \text{or } w_i \text{ contains any special symbol or digit} \end{cases}$
Pre	Pre$_i(n) = \begin{cases} \text{Prefix string of length } n \text{ of } w_i & \text{if } \lvert w_i \rvert \geq n \\ ND(=0) & \text{if } \lvert w_i \rvert \leq (n-1) \\ & \text{or } w_i \text{ is a punctuation symbol} \\ & \text{or } w_i \text{ contains any special symbol or digit} \end{cases}$
NE	$NE_i =$ NE tag of w_i
FirstWord	FirstWord$_i = \begin{cases} 1, \text{if } w_i \text{ is the first word of a sentence} \\ 0, \text{otherwise} \end{cases}$
CntDgt	CntDgt$_i = \begin{cases} 1, \text{if } w_i \text{ contains digit} \\ 0, \text{otherwise} \end{cases}$
FourDgt	FourDgt$_i = \begin{cases} 1, \text{if } w_i \text{ consists of four digits} \\ 0, \text{otherwise} \end{cases}$
TwoDgt	TwoDgt$_i = \begin{cases} 1, \text{if } w_i \text{ consists of two digits} \\ 0, \text{otherwise} \end{cases}$
CntDgtCma	CntDgtCma$_i = \begin{cases} 1, \text{if } w_i \text{ contains digit and comma} \\ 0, \text{otherwise} \end{cases}$
CntDgtPrd	CntDgtPrd$_i = \begin{cases} 1, \text{if } w_i \text{ contains digit and period} \\ 0, \text{otherwise} \end{cases}$
CntDgtSlsh	CntDgtSlsh$_i = \begin{cases} 1, \text{if } w_i \text{ contains digit and slash} \\ 0, \text{otherwise} \end{cases}$
CntDgtHph	CntDgtHph$_i = \begin{cases} 1, \text{if } w_i \text{ contains digit and hyphen} \\ 0, \text{otherwise} \end{cases}$
CntDgtPrctg	CntDgtPrctg$_i = \begin{cases} 1, \text{if } w_i \text{ contains digit} \\ \quad \text{and percentage} \\ 0, \text{otherwise} \end{cases}$
Infrequent	Infrequent$_i = I_{\{\text{Infrequent word list}\}}(w_i)$
Length	Length$_i = \begin{cases} 1, \text{if } w_i \geq 3 \\ 0, \text{otherwise} \end{cases}$

– Part of Speech information: For POS tagging, we have used a CRF-based POS tagger [59], which has been developed with the help of a tagset of 26 different POS tags [5], defined for the Indian languages. We have used the inflection lists that can appear with the different wordforms of noun, verb and adjectives, a lexicon [60] that has been developed in an unsupervised way from the Bengali news corpus, and the NE tags using a NER system [44] as the features of POS tagging in Bengali. This POS tagger has an accuracy of 90.2%.

The language dependent features are represented in Table 9.

5 Use of unlabeled data

We have developed two different techniques that use the large collection of unlabeled corpus [55] in NER. The first one is an unsupervised learning technique used to generate lexical context patterns for use as the features of the classifiers. The second one is a semi-supervised learning technique that is used to select the appropriate data from the large collection of documents. In the literature, unsupervised algorithms (bootstrapping from seed examples and unlabeled data) have been discussed in [61], [47], and [62]. Using a parsed corpus, the proper names that appear in certain syntactic contents were identified and classified in [61].The procedures to identify and classify proper names in seven languages, learning character-based contextual, internal, and morphological patterns are reported in [62]. This algorithm does not strictly require capitalization but recall was much lower for the languages that do not have case distinctions. Others such as [63] relied on structures such as appositives and compound nouns. Contextual patterns that predict the semantic class of the subject, direct object, or prepositional phrase object are reported in [64] and [65]. The technique to use the windows of tokens to learn contextual and internal patterns without parsing is described in [66] and [67]. The technique reported in [67] enable discovery of generalized names embedded in larger noun groups. An algorithm for unsupervised learning and semantic classification of names and terms is reported in [67]. They considered the *positive example* and *negative example* for a particular name class. We have developed an unsupervised algorithm that can generate the lexical context patterns from the unlabeled corpus. This work differs from the previous works in the sense that here we have also considered the patterns that yield *negative examples*. These *negative examples* can be effective to generate new patterns. Apart from *accuracy*, we have considered the *relative frequency* of a pattern in order to decide its inclusion into the final set of patterns. The final lexical context patterns have been used as features of the classifiers. Here, we have used a portion of the Bengali news corpus [55] that has been classified on geographic domain (International, National, State, District, Metro [Kolkata]) as well as on topic domain (Politics, Sports, Business). Statistics of this corpus is shown in Table 10.

[5] http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

5.1 Lexical context pattern learning

Lexical context patterns are generated from the unlabeled corpus of approximately 10 million wordforms, as shown in Table 10. Given a small seed examples and an unlabeled corpus, the algorithm can generate the lexical context patterns through bootstrapping. The seed name serves as a *positive example* for its own NE class, *negative example* for other NE classes and *error example* for non-NEs.

1. Seed list preparation: We have collected frequently occurring words from the Bengali news corpus and the annotated training set of 272K wordforms to use as the seeds. There are 123, 87, and 32 entries in the person, location, and organization seed lists, respectively.

2. Lexical pattern generation: The unlabeled corpus is tagged with the elements from the seed lists. For example,
<Person> *sonia gandhi* < /Person>, <Location> *kolkata* < /Location> and <Organization> *jadavpur viswavidyalya*< /Organization>.

   For each tag T inserted in the training corpus, the algorithm generates a lexical pattern p using a context window of maximum width 6 (excluding the tagged NE) around the left and the right tags, e.g., $p = [l_{-3}l_{-2}l_{-1} < T > \ldots < /T > l_{+1}l_{+2}l_{+3}]$, where, l_i are the context of p. Any of l_i may be a punctuation symbol. In such cases, the width of the lexical patterns will vary. We also generate the lexical context patterns by considering the left and right contexts of the labeled examples of the annotated corpus of 272K wordforms. All these patterns, derived from the different tags of the labeled and unlabeled training corpora, are stored in a Pattern Table (or, set P), which has four different fields namely, pattern *id* (identifies any particular pattern), pattern *example* (pattern), pattern *type* (*Person name*/*Location name*/*Organization name*) and *relative frequency* (indicates the number of times any pattern of a particular *type* appears in the entire training corpus relative to the total number of patterns generated of that *type*). This table has 38,198 entries, out of which 27,123 patterns are distinct. Labeled training data contributes to 15,488 patterns and the rest is generated from the unlabeled corpus.

3. Evaluation of patterns: Every pattern p in the set P is matched against the same unlabeled corpus. In a place, where the context of p matches, p predicts the occurrence of the left or right boundary of name. POS information of the words as well as some linguistic rules and/or length of the entity have been used in detecting the other boundary. The extracted entity may fall in one of the following categories:

Table 8: Gazetteers used in the experiment

Gazetteer	Number of entries	Source
NE suffix	115	Manually prepared
Organization suffix	94	Manually created from the news corpus
Person prefix	245	Manually created from the news corpus
Middle name	1491	Semi-automatically from the news corpus
Surname	5,288	Semi-automatically from the news corpus
Common Location	547	Manually developed
Action verb	221	Manually prepared
Designation words	947	Semi-automatically prepared from news corpus
First names	72,206	Semi-automatically prepared from the news corpus
Location name	5,125	Semi-automatically prepared from the news corpus
Organization name	2,225	Manually prepared
Month name	24	Manually prepared
Weekdays	14	Manually prepared
Measurement expressions	52	Manually prepared

Table 9: Descriptions of the language dependent features. Here, *i* represents the position of the current word and w_i represents the current word

Feature	Description
FirstName	$FirstName_i = I_{\{\text{First name list}\}}(w_i)$
MidName	$MidName_i = I_{\{\text{Middle name list}\}}(w_i)$
SurName	$SurName_i = I_{\{\text{Sur name list}\}}(w_i) \bigvee I_{\{\text{Sur name list}\}}(w_{i+1})$
Funct	$Funct_i = I_{\{\text{Function word list}\}}(w_i)$
MonthName	$MonthName_i = I_{\{\text{Month name list}\}}(w_i)$
WeekDay	$WeekDay_i = I_{\{\text{Week day list}\}}(w_i)$
MeasureMent	$Measurement_i = I_{\{\text{Measurement word list}\}}(w_{i+1})$ $\bigvee I_{\{\text{Measurement list}\}}(w_{i+1})$
POS	POS_i=POS tag of the current word
NESuf	$NESuf_i = I_{\{\text{NE suffix list}\}}(w_i)$
OrgSuf	$OrgSuf_i = I_{\{\text{Organization suffix word list}\}}(w_i)$ $\bigvee I_{\{\text{Organization suffix word list}\}}(w_{i+1})$
ComLoc	$ComLoc_i = I_{\{\text{Common location list}\}}(w_i)$
ActVerb	$ActVerb_i = I_{\{\text{Action verb list}\}}(w_i)$ $\bigvee I_{\{\text{Action verb ist}\}}(w_{i+1})$
DesG	$DesG_i = I_{\{\text{Designation word list}\}}(w_{i-1})$
PerPre	$PerPre_i = I_{\{\text{Person prefix word list}\}}(w_{i-1})$
LocName	$LocName_i = I_{\{\text{Location name list}\}}(w_i)$
OrgName	$OrgName_i = I_{\{\text{Organization name list}\}}(w_i)$

Table 10: Corpus statistics

Total number of news documents in the corpus	35, 143
Total number of sentences in the corpus	940, 927
Average number of sentences in a document	27
Total number of wordforms in the corpus	9, 998, 972
Average number of wordforms in a document	285
Total number of distinct wordforms in the corpus	152, 617

(a) *positive example*: The extracted entity is of the same NE type as that of the pattern.

(b) *negative example*: The extracted entity is of the different NE type as that of the pattern.

(c) *error example*: The extracted entity is not at all a NE.

4. Candidate pattern acquisition: For each pattern p, we have maintained three different lists for the *positive*, *negative* and *error* examples. The *type* of the extracted entity is determined by checking whether it appears in any of the seed lists (person/location/organization); otherwise, its *type* is determined manually. The *positive* and *negative* examples are then added to the appropriate seed lists. We then compute the pattern's *accuracy* as follows: $accuracy(p) = \frac{|positive(p)|}{[|positive(p)|+|negative(p)|+|error(p)|]}$

   A threshold value of *accuracy* has been chosen in order to discard the patterns below this threshold. A pattern is also discarded if its total *positive count* is less than a predetermined threshold value. The remaining patterns are ranked by their *relative frequency* values. The *n* top high frequent patterns are retained in the pattern set P and this set is denoted as *Accept Pattern*.

5. Generation of new patterns: All the positive and negative examples extracted by a pattern p in Step 4 can be used to generate further patterns from the same training corpus. Each new *positive* or *negative* instance (not appearing in the seed lists) is used to further tag the training corpus. We repeat steps 2-4 for each new NE until no new patterns can be generated. The threshold values of *accuracy, positive count* and *relative frequency* are chosen in such a way that in each iteration of the algorithm at least 5% new patterns are added to the set P. A newly generated pattern may be identical to a pattern that is already in the set P. In such a case, the *type* and *relative frequency* fields in the set P are updated accordingly. Otherwise, the newly generated pattern is added to the set with the *type* and *relative frequency* fields set properly. The algorithm terminates after 23 iterations and there are 34,298 distinct entries in the set P.

5.2 Unlabeled document and sentence selection using bootstrapping

We have divided the unlabeled 35,143 news documents based on news sources/types, i.e., International, National, State, District, Metro [Kolkata], Politics, Sports, Business etc. in order to create segments of manageable size. This helps us to separately evaluate the contribution of each segment using a gold standard development test set and reject those that are not helpful and to apply the latest updated best model to each subsequent segment. We have observed that the use of unlabeled data becomes effective if it is related to the target problem, i.e., the test set. So, appropriate unlabeled document selection is very essential. After selecting the documents, it is necessary to select the tagged sentences that are useful to improve the system performance. Appropriate sentences are selected based on majority voting and depending upon the structure and/or the contents of the sentences.

– Unlabeled Document Selection: The unlabeled data supports the acquisition of new names and contexts to provide new evidences to be incorporated in ME, CRF and SVM classifiers. Old estimates of the models may be worsened by the unlabeled data if it adds too many names whose tags are incorrect, or at least are incorrect in the context of the labeled training data and the test data. Unlabeled data can degrade rather than improve the classifier's performance on the test set if it is irrelevant to the test document. So, it is necessary to measure the relevance of the unlabeled data to our target test set.

  We construct a set of key words from the test set T to check whether unlabeled document d is useful or not.

  – We do not use all the words in test set T as the key words since we are only concerned about the distribution of name candidates. So, each document is tested with the CRF model that is developed with the language independent features (i.e, *baseline*), context features and gazetteers.

  – It is insufficient to take only the name candidates in the top one hypothesis for each sentence.

  Thus, we take all the name candidates in the top N best hypotheses ($N = 10$) for each sentence of the test set T to construct a query set Q. Using this query

set, we find all the relevant documents that include three (heuristically set) names belonging to the set Q. In addition, the documents are not considered if they contain fewer than seven (heuristic) names.

– Sentence Selection: All the tagged sentences of a relevant document are not added to training corpus as incorrectly tagged or irrelevant sentences can lead to the degradation in model performance. We are actually concerned on how much new information is extracted from each sentence of the unlabeled data compared to the training corpus that already we have in our hand.

We have used majority voting approach to select the relevant sentences. All the relevant documents are tagged with the ME, CRF, SVM-F and SVM-B models. If the majority of models agree to the same output for at least 80% of the words in a sentence then that sentence is selected to be added to the training corpus. This criterion often selects some sentences which are too short or do not include any name. These words may make the model worse if added to the training data. For example, the distribution of non-names may increase significantly leading to degradation of model performance. In this experiment, we have not included the sentences that include fewer than five words or do not include any names.

The bootstrapping procedure is given as follows:

1. Select a relevant document *RelatedD* from a large corpus of unlabeled data with respect to the test set T using the document selection method described earlier.
2. Split *RelatedD* into n subsets and mark them $C_1, C_2, \ldots, C_n$.
3. Call the development set *DevT*.
4. For $I = 1$ to n
   (a) Run initial ME, CRF, SVM-F and SVM-B on C_i.
   (b) For each tagged sentence S in C_i , if at least 80% of the words agree with the same outputs by the majority of models then keep S; otherwise, remove S.
   (c) Assign outputs to the remaining words from the SVM-F model.
   (d) If the length of S is less than five words or it does not contain any name then discard S.
   (e) Add C_i to the training data and retrain each model. This produces the updated models.
   (f) Run the updated models on *DevT*; if the performance gets reduced then do not use C_i and use the old models.
5. Repeat steps 1-4 until performance of each model becomes identical in two consecutive iterations.

Table 11: Statistics of the training, development and test sets

	Training	Development	Test
# of sentences	21,340	3,367	2,501
#of wordforms	272,000	50,000	35,000
#of NEs	22,488	3,665	3,178
#Avg. length of NE	1.5138	1.6341	1.6202

6 Evaluation results and discussions

We have manually annotated approximately 200K wordforms of the Bengali news corpus [55] with *Person name*, *Location name*, *Organization name* and *Miscellaneous name* NE tags with the help of *Sanchay Editor* [6] , a text editor for the Indian languages. Out of 200K wordforms, 150K wordforms along with the IJCNLP-08 shared task data has been used for training the models. Out of 200K wordforms, 50K wordforms have been used as the development data. The system has been tested with a gold standard test set of 35K wordforms. Statistics of the training, development and test sets are given in Table 11.

A number of experiments have been carried out taking the different combinations of the available words, context and orthographic word level features to identify the best-suited set of features in the ME, CRF and SVM frameworks for NER in Bengali. Evaluation results of the development set for the *baseline* models are presented in Table 12. The *baseline* ME based system performs best for the context word window of size three, dynamic NE tag of the previous word, suffixes and prefixes of length upto three characters of the current word, POS tag of the current word and other word-level language independent features. The system has demonstrated the overall f-score value of 72.49%. The *baseline* CRF model has shown best performance with the f-score of 75.71% for the context window of size five, dynamic NE information of the previous word, POS information of the current and previous words, prefixes and suffixes of length upto three characters of the current word along with other features. The SVM-F based *baseline* system has performed best among the three models and has demonstrated the f-score value of 76.3% for the context window of size six, NE information of the previous two words, POS information of the current, previous and the next words along with the other set of features as like CRF. The SVM-B has shown the f-score value of 76.1% with the same set of features used in SVM-F. In SVM models, we have conducted experiments with the different *polynomial kernel* functions and observed the highest f-score value with degree 2.

The language dependent features as described in Table 9 are included into the *baseline* models and the evaluation results are reported in Table 13. We have observed that all the gazetteers are not equally important to improve the per-

[6] Sourceforge.net/project/nlp-sanchay

Table 12: Results of the *baseline* models

Model	R (in %)	P (in %)	FS (in %)
ME	73.55	71.45	72.49
CRF	75.97	75.45	75.71
SVM-F	77.14	75.48	**76.30**
SVM-B	77.09	75.14	76.10

Table 13: Results including language dependent features

Model	R (in %)	P (in %)	FS (in %)
ME	75.26	74.91	74.41
CRF	79.03	80.62	79.82
SVM-F	81.37	80.14	**80.75**
SVM-B	81.29	79.16	80.21

formance of the classifiers. The use of gazetteers increases the performance by 2.43%, 4.11%, 4.45%, and 4.11% in the ME, CRF, SVM-F, and SVM-B classifiers, respectively. Results show that the effect of language dependent features is not very impressive in ME model. Thus, it can be decided that the use of all available features can not always improve the performance in a ME model and careful feature selection is very important.

6.1 Use of context patterns as features

High ranked patterns in the *Accept Pattern set* can be used as the features of the individual classifier. Words in the left and/or the right contexts of person, location and organization names carry effective information that could be helpful in their identification. A feature 'ContextInformation' is defined by observing the words in the window $[-3, 3]$ (three words spanning to left and right) of the current word in the following way:
•Feature value is 1 if the window contains any word of the pattern type *Person name*.
• Feature value is 2 if the window contains any word of the pattern type *Location name*.
•Feature value is 3 if the window contains any word of the pattern type *Organization name*.
• Feature value is 4 if the window contains any word that appears with more than one type.
• Feature value is 0 for those if the window does not contain any word of any pattern.

Experimental results of the system for the development set are presented in Table 14 by including the context features. Evaluation results show the effectiveness of context features with the improvement of f-scores by **3.17%**, **3.08%, 2.82%**, and **3.28%** in the ME, CRF, SVM-F, and SVM-B models, respectively. So, the context features are effective in improving the performance of all the models.

Table 14: Results using context features

Model	R (in %)	P (in %)	FS (in %)
ME	78.26	76.91	77.58
CRF	82.07	83.75	82.90
SVM-F	84.56	82.60	83.57
SVM-B	84.42	82.58	83.49

6.2 Post-processing techniques

We have conducted error analysis for all the classifiers with the help of confusion matrices. Several post-processing techniques have been adopted in order to improve the performance of each of the classifiers. It has been observed that the SVM models have the highest tendency of assigning NE tags to the words that are actually not NEs. In ME model, a lot of NEs are not identified at all. CRF model also suffers from this problem. The most confusing pairs of classes in these two models are LOC vs NNE, MISC vs NNE, PER vs NNE, E-ORG vs NNE and B-MISC vs MISC. On the other hand the most confusing pairs are LOC vs NNE, PER vs NNE, MISC vs NNE and E-ORG vs NNE. Depending upon the errors involved in the models, we have developed various mechanisms to improve the recall and precision values of the classifiers.

– Class decomposition technique for SVM: Unlike CRF, SVM model does not predict the NE tags to the constituent words depending upon the sentence. SVM predicts the class depending upon the labeled word examples only. If target classes are equally distributed, the *pairwise* method can reduce the training cost. Here, we have a very unlabeled class distribution with a large number of samples belonging to the class 'NNE' (other than NEs) (Table 11). This leads to the same situation like *one-vs-rest* strategy. One solution to this unbalanced class distribution is to decompose the 'NNE' class into several subclasses effectively. Here, we have decomposed the 'NNE' class according to the POS information of the word. That is, given a POS tagset POS, we produce new $|POS|$ classes, '$NNE - C'|C \in POS$. So, we have 26 subclasses which correspond to non-NE regions such as 'NNE-NN' (common noun), 'NNE-VFM' (verb finite main) etc. Experimental results have shown the recall, precision, and f-score values of 87.09%, 86.73%, and 86.91%, respectively, in the SVM-F model and 87.03%, 85.98%, and 86.5%, respectively, in SVM-B model. We have also conducted similar experiments in the CRF models and observed the lower f-score values.

– Post-processing with the n-best outputs for CRF: There are inconsistent results in the CRF model in some cases. We have performed a post-processing step to correct these errors. The post-processing tries to assign the correct tag according to the n-best results

for every sentence of the test set. We have considered the top 15 labeled sequences for each sentence with the confidence scores. Initially, we collect the NEs from the high confident results and then we re-assign the tags for low confident results using this NE list. The procedure is given below: S is the set of sentences in the test set, i.e, $S = \{s_1, s_2, \ldots, s_n\}$; R is set of n-best result ($n = 15$) of S, i.e, $R = \{r_1, r_2, \ldots, r_n\}$, where r_i is a set of n-best results of s_i; c_{ij} is the confidence score of r_{ij}, that is the jth result in r_i.

Creation of NE set from the high confident tags:

for $i = 1$ to n {if $(r_{i0} \geq 0.6)$ then collect all NEs from r_{i0} and add to the set NESet }.
Replacement:
for $i = 1$ to n {if $(r_{i0} \geq 0.6)$ then Result(s_i)=r_{i0} ;
else { TempResult(s_i)=r_{i0} ;
for $j = 1$ to m {if (NEs of r_{ij} are included in NESet) then Replace the NE tags
of TempResult with these new tags}.
Result(s_i) =TempResult(s_i)}}.

Evaluation results have demonstrated the recall, precision, and f-score values of 86.75%, 85.91%, and 86.33%, respectively, in the CRF model. Thus, there is an improvement of **4.43%** f-score in the CRF model.

– Post-processing the output of ME model: We have used the following heuristics to further improve the performance of the ME model. Some of the rules are useful to improve the recall values, whereas some are effective to increase the precisions. Many of the heuristics are also helpful to identify the boundaries properly. Following are the set of heuristics.

1. The NNE tag of a particular word is replaced by the appropriate NE tag, if that word appears somewhere in the output with that NE.
2. If any word is tagged as B-XXX/I-XXX/E-XXX (XXX: PER/LOC/ORG/MISC) and the previous and next words are tagged as NNE then that word is assigned the NE tag of type XXX.
3. The NNE tag of a word is replaced by the E-XXX if the previous word is already tagged as B-XXX.
4. NNE tag of a word is replaced by B-XXX, if the next word is already tagged as E-XXX.
5. If there is sequence B-XXX/I-XXX followed by XXX in the output, then the tag XXX is replaced by the E-XXX.
6. If the sequence of tags is of the form XXX B-XXX1/I-XXX1/E-XXX1 NNE (XXX#XXX1) for three consecutive words in the output, then the tag B-XXX1/I-XXX1/E-XXX1 is replaced by the XXX1.
7. If current word is not tagged as B-XXX/I-XXX/NNE but the following word is tagged as B-XXX/I-XXX/E-XXX then the current word is assigned the tag B-XXX.
8. If the words, tagged as NNE, contain the variable length NE suffixes (used as the feature in the *baseline* models) then the words are assigned the NE tags. The types of the NE tags are determined by the types of the suffixes (e.g., Person tag is assigned if matches with the person name suffix).

Evaluation results have demonstrated the recall, precision, and f-score values of 81.55%, 78.67%, and 80.8%, respectively.

6.3 Impact of unlabeled data selection

In order to investigate the contribution of document selection in bootstrapping, we run the post-processed models on 35,143 news documents. This yields the gradually improving performance for the models as shown in Table 15.

We have also carried out experiments with the same unlabeled data in order to observe the effectiveness of document selection and sentence selection separately. Results are reported in Table 16. Row 2 of the table represents results of the post-processed models that are used to tag the unlabeled documents to be included into the initial training set in a bootstrapped manner. This presents the results by using the majority voting selection criterion only. Comparing row 2 with row 3, we find that not using document selection, even though it multiplies the size of the training corpus, results in 1.04%, 1.36%, 1.02%, and 0.83% lower performance in the ME, CRF, SVM-B, and SVM-F models, respectively. This leads us to conclude that simply relying upon large corpus is not in itself sufficient. Effective use of large corpus demands good selection criterion of documents to remove off-topic materials. The system has demonstrated the f-score values of 83.87%, 89.34%, 89.55%, and 89.37% in the ME, CRF, SVM-F, and SVM-B models, respectively, by adding the sentence selection method.

6.4 Voting techniques

Voting scheme is effective in order to improve the overall performance of any multi-engine system. Here, we have combined four models using three different voting mechanisms. But before applying weighted voting, we need to decide the weights to be given to the individual system. We can obtain the best weights if we could obtain the accuracy for the 'true' test data. However, it is impossible to estimate them. Thus, we have used following weighting methods in our experiments:

1. Uniform weights (Majority voting): We have assigned the same voting weight to all the systems. The com-

Table 15: Incremental improvement of performance

Iteration	Sentences added	FS (in %)			
		ME	CRF	SVM-F	SVM-B
0	0	80.8	86.33	86.91	86.5
1	107	81.2	86.9	87.27	87.13
2	213	81.67	87.35	87.53	87.41
3	311	81.94	87.93	88.12	87.99
4	398	82.32	88.11	88.25	88.18
5	469	82.78	88.66	88.83	88.71
6	563	82.94	89.03	89.17	89.08
7	619	83.56	89.12	89.27	89.15
8	664	83.79	89.28	89.35	89.22
9	691	83.85	89.34	89.51	89.37
10	701	83.87	89.34	89.55	89.37
11	722	83.87	89.34	89.55	89.37

Table 16: Incremental improvement of performance

Model		ME	CRF	SVM-F	SVM-B
1	Post-processed	80.8	86.33	86.91	86.50
2	(1)+ Bootstrapping	82.01	87.36	88.05	87.81
3	(2) + Document selection	83.05	88.72	88.88	88.83
4	(3) + Sentence selection	83.87	89.34	89.55	89.37

bined system selects the classifications, which are proposed by the majority of the models. If four outputs are different, then the output of the SVM-F system is selected.

2. Cross validation f-score values: The training data is divided into N portions. We employ the training by using $N - 1$ portions, and then evaluate the remaining portion. This is repeated N times. In each iteration, we have evaluated the individual system following the similar methodology, i.e., by including the various gazetteers and the same set of post-processing techniques. At the end, we get N f-score values for each of the system. Final voting weight for a system is given by the average of these N f-score values. Here, we set the value of N to be 10. We have defined two different types of weights depending on the cross validation f-score as follows:

   – Total F-Score: In the first method, we have assigned the overall average f-score of any classifier as the weight for it.

   – Tag F-Score: In the second method, we have assigned the average f-score value of the individual tag as the weight for that model.

Experimental results of the voted system are presented in Table 17. Evaluation results show that the system achieves the highest performance for the voting scheme 'Tag F-Score'. Voting shows (Tables 16-17) an overall improvement of **8.84%** over the least performing ME based system and **3.16%** over the best performing SVM-F system in terms of f-score values.

Table 17: Results of the voted system (development set)

Voting	R (in %)	P (in %)	FS (in %)
Majority	93.19	89.35	91.23
Total F-Score	93.85	89.97	92.17
Tag F-Score	93.98	91.46	92.71

6.5 Experimental results of the test set

The systems have been tested with a gold standard test set of 35K wordforms. Approximately, 25% of the NEs are unknown in the test set. Experimental results of the test set for the *baseline* models have shown the f-score values of 73.15%, 76.35%, 77.36%, and 77.23% in the ME, CRF, SVM-F, and SVM-B based systems, respectively. Results have demonstrated the improvement in f-scores by 8.35%, 9.67%, 8.82% and 8.83% in the ME, CRF, SVM-B, and SVM-F models, respectively, by including the language specific features, context features and post-processing techniques. Appropriate unlabeled sentences are then selected by the document and sentence selection methods to be included into the training set. Models have shown the f-scores of 83.77%, 89.02%, 89.17%, and 89.11% in the ME, CRF, SVM-F, and SVM-B models, respectively. Experi-

Table 18: Results of the voted system (test set)

Voting	R (in %)	P (in %)	FS (in %)
Majority	92.91	89.77	91.31
Total F-Score	93.55	90.16	91.82
Tag F-Score	93.79	91.34	92.55

Table 19: Comparison with other Bengali NER systems

Model	R(%)	P (%)	FS (%)
A ([40])	66.53	63.45	64.95
B ([40])	69.32	65.11	67.15
HMM ([41])	74.02	72.55	73.28
ME ([42])	78.64	76.89	77.75
CRF ([43])	80.02	80.21	80.15
SVM ([68])	81.57	79.05	80.29
Proposed system	93.79	91.34	92.55

mental results of the voted system are presented in Table 18. Results show that the voting scheme that considers the f-score value of the individual NE tag as the weight of a particular classifier, i.e., 'Tag F-Score' gives the best result among the three voting methods. The voted system has demonstrated the improvement in the f-scores by 8.78%, 3.53%, 3.38%, 3.44%, in the ME, CRF, SVM-F, and SVM-B systems, respectively.

The existing Bengali NER systems based on the pattern directed shallow parsing approach[40], HMM [41], ME [59], CRF [43], and SVM [68] have been evaluated with the same datasets. Comparative evaluation results are presented in Table 19. Comparisons with the works reported in the IJCNLP-08 shared task are out of scope because of the following reasons:

- The shared task was involved with a fine-grained tagset of twelve NE tags. In this work, we have considered only the tags that denote person name, location name, organization name, date, time and number expressions.

- The main challenge of the shared task was to identify and classify the nested NEs (i.e, the constituent parts of a bigger NE). Here, we are not concerned with the nested NEs.

Results show the effectiveness of the proposed NER system that outperforms other existing systems by the impressive margins. Thus, it can be decided that contextual information of the words, several post-processing methods and the use of appropriate unlabeled data can yield a reasonably good performance. Results also suggest that combination of several classifiers is more effective than any single classifier.

7 Conclusion

In this paper, we have reported a NER system by combining the classifiers, namely ME, CRF and SVM with the help of weighted voting techniques. We have manually annotated a portion of the Bengali news corpus, developed from the web archive of a leading Bengali newspaper. In addition, we have also used the IJCNLP-08 NER shared task data tagged with a fine-grained NE tag set of twelve tags. We have converted this data with the NE tags denoting person name, location name, organization name and miscellaneous name. The individual models make use of the different contextual information of the words, several orthographic word-level features and the binary valued features extracted from the various gazetteers that are helpful to predict the NE classes. A number of features are language independent in nature. We have used an unsupervised learning technique to generate lexical context patterns to be used as features of the classifiers. We have described the method of selecting appropriate unlabeled documents and sentences from a large collection of unlabeled data. This eliminates the necessity of manual annotation for preparing the NE annotated corpus. We have also shown how several heuristics for ME, n-best output of CRF and the class splitting technique of SVM are effective in improving the performance of the corresponding model. Finally, the outputs of the classifiers have been combined with the three different weighted voting techniques. It has been shown that combination of several models performs better than any single one.

References

[1] Cunningham, H.: GATE, a General Architecture for Text Engineering. Computers and the Humanities **36** (2002) 223–254

[2] Babych, B., Hartley, A.: Improving Machine Translation Quality with Automatic Named Entity Recognition. In: Proceedings of EAMT/EACL 2003 Workshop on MT and other Language Technology Tools. (2003) 1–8

[3] Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A., Bolohan, O.: LCC Tools for Question Answering. In: Text REtrieval Conference (TREC) 2002. (2002)

[4] Y.C. Wu, T.K. Fan, Y.L., Yen, S.: Extracting Named Entities using Support Vector Machines. In: Springer-Verlag. (2006)

[5] I. Budi, S.B.: Association Rules Mining for Name Entity Recognition. In: Proceedings of the Fourth International Conference on Web Information Systems Engineering. (2003)

[6] Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D.J., Tyson, M.: Fastus: A finite-state processor for information extraction from real-world text. In: IJCAI. (1993) 1172–1178

[7] Appelt, D.E., Hobbs, J.R., Bear, J., Israel, D., Kameyama, M., Kehler, A., Martín, D., Myers, K., Tyson, M. Proceedings of the 6th Messsage Understanding Conference. In: SRI International FASTUS System MUC-6 Test Results and Analysis. Morgan Kaufmann Publishers, Inc., Columbia, Maryland (1995) 237–248

[8] Iwanska, L., Croll, M., Yoon, T., Adams, M.: Wayne State University: Description of the UNO Processing System as used for MUC-6Language Independent Named Entity Recognition. In: Proceedings of the MUC-6, Morgan-Kauffman Publisher (1995)

[9] Grishman, R.: The NYU System for MUC-6 or Where's the Syntax. In: Proceedings of the MUC-6, Morgan-Kauffman Publisher (1995)

[10] Farmakiotou D., Karkaletsis V., K.J.S.G.S.C., P., S.: Rule-based Named Entity Recognition for Greek Financial Text. In: Proceedings of the Workshop on Computational Lexicography and Multimedia Dictionaries (COMLEX 2000). (2000) 75–78

[11] Wolinski F., V.F., B., D.: Automatic Processing of Proper Names in Texts. In: In Proceedings of the European Chapter of the Association for Computer Linguistics (EACL), Dublin, Ireland (1995) 23–30

[12] Wolinski F., V.F., M., S.: Using Learning-based Filters to Detect Rule-based Filtering Obsolescence. In: In Recherche d' Information Assistee par Ordinateur (RIAO), Paris, France (2000) 1208–1220

[13] Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunnigham, H., Wilks, Y.: Univ. Of Sheffield: Description of the LaSIE-II System as Used for MUC-7. In: MUC-7, Fairfax, Virginia (1998)

[14] Aone, C., Halverson, L., Hampton, T., Ramos-Santacruz, M.: SRA: Description of the IE2 system used for MUC-7. In: MUC-7, Fairfax, Virginia (1998)

[15] Mikheev, A., Grover, C., Moens, M.: Description of the LTG system used for MUC-7. In: MUC-7, Fairfax, Virginia (1998)

[16] Mikheev, A., Grover, C., Moens, M.: Named Entity Recognition without Gazeteers. In: Proceedings of EACL, Bergen, Norway (1999) 1–8

[17] Zhou, G., Su, J.: Named Entity Recognition using an HMM-based Chunk Tagger. In: Proceedings of ACL, Philadelphia (2002) 473–480

[18] Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schawartz, R., Stone, R., Weischedel, R., the Annotation Group: BBN: Description of the SIFT System as Used for MUC-7. In: MUC-7, Fairfax, Virginia (1998)

[19] Bikel, D.M., Schwartz, R.L., Weischedel, R.M.: An Algorithm that Learns What's in a Name. Machine Learning **34** (1999) 211–231

[20] Borthwick, A.: Maximum Entropy Approach to Named Entity Recognition. PhD thesis, New York University (1999)

[21] Borthwick, A., Sterling, J., Agichtein, E., Grishman, R.: NYU:Description of the MENE Named Entity System as Used in MUC-7. In: MUC-7, Fairfax (1998)

[22] Sekine, S.: Description of the Japanese NE System used for MET-2. In: MUC-7, Fairfax, Virginia (1998)

[23] Bennet, S.W., Aone, C., Lovell, C.: Learning to Tag Multilingual Texts Through Observation. In: Proceedings of Empirical Methods of Natural Language Processing, Providence, Rhode Island (1997) 109–116

[24] McCallum, A., Li, W.: Early results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-enhanced Lexicons. In: Proceedings of CoNLL, Canada (2003) 188–191

[25] Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: ICML. (2001) 282–289

[26] Yamada, H., Kudo, T., Matsumoto, Y.: Japanese Named Entity Extraction using Support Vector Machine. In Transactions of IPSJ **43** (2001) 44–53

[27] Kudo, T., Matsumoto, Y.: Chunking with Support Vector Machines. In: Proceed-ings of NAACL. (2001) 192–199

[28] Takeuchi, K., Collier, N.: Use of Support Vector Machines in Extended Named Entity Recognition. In: Proceedings of the 6th Conference on Natural Language Learning, (CoNLL-2002). (2002) 119–125

[29] Masayuki, A., Matsumoto, Y.: Japanese Named Entity Extraction with Redundant Morphological Analysis. In: NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Morristown, NJ, USA, Association for Computational Linguistics (2003) 8–15

[30] Collins, M., Singer, Y.: 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)

[31] J. Kim, I.K., Choi, K.: Unsupervised Named Entity Classification Models and their Ensembles. In: Proceedings of the 19th Conference on Computational Linguistics. (2002)

[32] A. Mikheev, C.G., Moens, M.: Description of the LTG System for MUC-7. In: Proceedings of Message Understanding Conference (MUC-7). (1998)

[33] R. Srihari, C.N., Li, W.: A Hybrid Approach for Named Entity and Sub-type Tagging. In: Proceedings of Sixth Conference on Applied Natural Language Processing. (2000) 247–254

[34] Wu, D., Ngai, G., Carpuat, M.: A stacked, voted, stacked model for named entity recognition. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, Morristown, NJ, USA, Association for Computational Linguistics (2003) 200–203

[35] Florian, R., Ittycheriah, A., Jing, H., Zhang, T.: Named entity recognition through classifier combination. In: Proceedings of CoNLL-2003, Edmonton, Canada (2003) 168–171

[36] Miller, S., Guinness, J., Zamanian, A.: Name tagging with word clusters and discrimina-tive training. In: Proc of HLT-NAACL 2004, Boston, USA (2004) 337–342

[37] Lin, W., Yangarber, R., Grishman, R.: Bootstrapped learning of semantic classes from positive and negative examples. In: In Proceedings of ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data. (2003) 103–111

[38] Bean, D., Rilof, E.: Unsupervised learning of contextual role knowledge for coreference resolution. In: Proc. of HLT-NAACL 2004. (2004) 297–304

[39] Ji, H., Grishman, R.: Data selection in semi-supervised learning for name tagging. In: Proc. of the Workshop on Information Extraction Beyond the Document. (2006)

[40] Ekbal, A., Bandyopadhyay, S.: Lexical Pattern Learning from Corpus Data for Named Entity Recognition. In: Proceedings of ICON, India (2007) 123–128

[41] Ekbal, A., Naskar, S., Bandyopadhyay, S.: Named Entity Recognition and Transliteration in Bengali. Named Entities: Recognition, Classification and Use, Special Issue of Lingvisticae Investigationes Journal **30** (2007) 95–114

[42] Ekbal, A., Bandyopadhyay, S.: Maximum entropy approach for named entity recognition in bengali. In: Proceedings of the 7th International Symposium on Natural Language Processing (SNLP-08), Thailand (2007)

[43] Ekbal, A., R.Haque, Bandyopadhyay, S.: Named Entity Recognition in Bengali: A Conditional Random Field Approach . In: Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP 2008). (2008) 589–594

[44] Ekbal, A., Bandyopadhyay, S.: Bengali Named Entity Recognition using Support Vector Machine. In: Proceedings of Workshop on NER for South and South East Asian Languages, 3rd International Joint Conference on Natural Languge Processing (IJCNLP), India (2008) 51–58

[45] Ekbal, A., Bandyopadhyay, S.: Appropriate Unlabeled Data, Post-processing and Voting can Improve the Performance of a NER System. In: Proceedings of the 6th International Conference on Natural Language Processing (ICON), Pune, India, Macmillan Publishers (2008) 234–239

[46] Li, W., McCallum, A.: Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction. ACM Transactions on Asian Languages Information Processing **2** (2004) 290–294

[47] Cucerzon, S., Yarowsky, D.: Language independent named entity recognition combining morphological and contextual evidence. In: Proceedings of the 1999 Joint SIGDAT conference on EMNLP and VLC, Washington, D.C. (1999)

[48] Saha, S., Sarkar, S., Mitra, P.: A Hybrid Feature set based Maximum Entropy Hindi Named Entiy Recognition. In: Proceedings of the 3rd International Joint Conference in Natural Langauge Processing (IJCNLP 2008). (2008) 343–350

[49] Kumar, N., Bhattacharyya, P.: Named entity recognition in hindi using memm. Technical report, IIT Bombay, India (2006)

[50] Gali, K., Sharma, H., Vaidya, A., Shisthla, P., Sharma, D.M.: Aggregrating Machine Learning and Rule-based Heuristics for Named Entity Recognition. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages. (2008) 25–32

[51] Kumar, P.P., Kiran, V.R.: A Hybrid Named Entity Recognition System for South Asian Languages. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages. (2008) 83–88

[52] Srikanth, P., Murthy, K.N.: Named Entity Recognition for Telugu. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages. (2008) 41–50

[53] Vijayakrishna, R., Sobha, L.: Domain Focused Named Entity Recognizer for Tamil using Conditional Random Fields. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages. (2008) 93–100

[54] Shishtla, P.M., Pingali, P., Varma, V.: A Character n-gram Based Approach for Improved Recall in Indian Language ner. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages. (2008) 101–108

[55] Ekbal, A., Bandyopadhyay, S.: A Web-based Bengali News Corpus for Named Entity Recognition. Language Resources and Evaluation Journal **42** (2008) 173–182

[56] Bharati, A., R.S., Sharma, D.M.: Shakti Analyser: SSF Representation. In: http://shiva.iiit.ac.in/SPSAL2007/ssf-analysisrepresentation.pdf. (2005)

[57] Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc., New York, NY, USA (1995)

[58] Brants, T.: TnT a Statistical Parts-of-Speech Tagger. In: Proceedings of the Sixth Conference on Applied Natural Language Processing ANLP-2000. (2000) 224–231

[59] Ekbal, A., Haque, R., Bandyopadhyay, S.: Bengali Part of Speech Tagging using Conditional Random Field. In: Proceedings of Seventh International Symposium on Natural Language Processing, SNLP2007, Thailand (2007)

[60] Ekbal, A., Bandyopadhyay, S.: Web-based bengali News Corpus for Lexicon Development and POS Tagging. Polibits (ISSN 1870 9044) **37** (2008) 20–29

[61] Collins, M., Singer, Y.: 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)

[62] Cucerzan, S., Yarowsky, D.: Language independent NER using a unified model of internal and contextual evidence. In: Proceedings of CoNLL 2002. (2002) 171–175

[63] Phillips, W., Riloff, E.: Exploiting strong syntactic heuristics and co-training to learn semantic lexicons. In: EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Morristown, NJ, USA, Association for Computational Linguistics (2002) 125–132

[64] Riloff, E., Jones, R.: Learning dictionaries for information extraction by multi-level bootstrapping. In: AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence, Menlo Park, CA, USA, American Association for Artificial Intelligence (1999) 474–479

[65] Thelen, M., Riloff, E.: A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In: EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Morristown, NJ, USA, Association for Computational Linguistics (2002) 214–221

[66] Strzalkowski, T., Wang, J.: A self-learning universal concept spotter. In: Proceedings of the 16th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1996) 931–936

[67] Yangarber, R., Lin, W., Grishman, R.: Unsupervised learning of generalized names. In: Proceedings of the 19th international conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2002) 1–7

[68] Ekbal, A., Bandyopadhyay, S.: Bengali Named Entity Recognition using Support Vector Machine. In: Proceedings of NERSSEAL, IJCNLP-08. (2008) 51–58