Named Entity Recognition, Classification and Transliteration in Bengali

Asif Ekbal

Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India

and

EMMA Post-doctoral Student, Department of Computational Linguistics, University of Heidelberg, Germany

Email: ekbal@cl.uni-heidelberg.de
asif.ekbal@gmail.com

Contents

Contents

Motivation

- India is a multilingual country with great cultural diversities
- Languages of India belong to the following groups
  - Indo-European family<--- Old Indo-Aryan family (e.g., Sanskrit) (70% speakers)→Northern India (e.g., Gujarati, Hindi, Marathi, Saraiki, Punjabi, Sindhi, Bengali, Oriya etc.)
  - Dravidian family (22% speakers) →Southern India (e.g., Tamil, Telugu, Kannada, Malayalam etc.)
  - Austro-Asiatic family
  - Tibeto-Burman
  - Language-isolates

North-Eastern India (e.g., Bodo, Manipuri etc.)

8%

Nihali language (Tribal area of India)

- Number of official languages in India: 22 (8^{th} Schedule)

Motivation (Contd..)

- Bengali
  - Emerged in AD 1000
  - Spoken in West Bengal, Tripura, Assam and Jharkhand states of India (Rank 2 in India)
  - National language of Bangladesh
  - Rank 7^{th} in the World in terms of native speakers

- NERC in Indian languages
  - More difficult and challenging
  - Efforts are still in infancy
  - Only available works in Indian languages when we started working →Cucerzon and Yarowsky, 1999; Li and McCallum, 2004
- NERC in Bengali
  - No available works when we started
  - We initiated the works !

- Appropriate approach for NERC for a less computerized language

- Resource constrained nature of the language

Motivation (Contd..)

- Another important motivation was to create sufficiently large Bengali corpus, NE tagged data, gazetteers, POS taggers, bilingual dictionaries etc. for NERC, Transliteration as well as for other application areas

- NE Transliteration in Indian languages
  - No available work when we started
  - We initiated the works !

- Importance of NE transliteration in a multilingual country like India
  - Large collections of *person names*, *location names* and o*rganization names* like census data, electoral roll and railway reservation information must be available to citizens of the country in their own vernacular

- Orthographic transliteration framework rather than conventional phoneme-based framework

- To propose a generalized transliteration algorithm, applicable for any language pair of comparable orthography (e.g., English and other Indian languages)

What is Named Entity Recognition and Classification (NERC)?

- NERC – Named Entity Recognition and Classification (NERC) involves identification of proper names in texts, and classification into a set of pre-defined categories of interest as:

  - Person names (names of people)

  - Organization names (companies, government organizations, committees, etc.)

  - Location names (cities, countries etc)

  - Miscellaneous names (Date, time, number, percentage, monetary expressions, number expressions and measurement expressions)

Approaches for NERC

- Broad Categories
  - Rule based NERC
  - Machine learning (ML) based NERC
    - Supervised ML technique
    - Semi-supervised ML technique
    - Unsupervised ML technique
  - Hybrid NERC
- Our Approaches
  - Active Learning Technique
  - Supervised ML Technique
    - Hidden Markov Model
    - Maximum Entropy
    - Conditional Random Field
    - Support Vector Machine
  - Semi-supervised ML Technique
  - Multi-Engine Approach based on Voting

Application areas of NERC

- Machine Translation
- Information Retrieval
- Question-Answering system
- Automatic Summarization

Named Entity (NE) Transliteration

- What is Transliteration?
  - Translating from one to another language by expressing the original foreign word using characters of the target language preserving the pronunciation in their source language
- Problem of NE Transliteration
  - Technical terms and NEs constitute the bulk of the Out Of Vocabulary (OOV) words
  - NEs usually not found in bilingual dictionaries and very generative in nature
  - NE transliteration → A tricky task (Translation and Transliteration both)

Example 1: জনতা দল (*janatA dal*)→*Janata Dal* (literal translation) (people party!)

জনতা (*janatA*)→people

দল (*dal*) →party

Vocabulary words

Example 2: যাদবপুর বিশ্ববিদ্যালয় (*yAdabpur bishvabidyAlaYa*)→*Jadavpur University*

যাদবপুর (*yAdabpur*)→ *Jadavpur* [Transliteration]

বিশ্ববিদ্যালয় (*bishvabidyAlaYa*)→*University* [Translation]

Named Entity Transliteration (Contd..)

- Two viewpoints of NE transliteration
  - Transliteration framework
    - Phoneme-based transliteration (Knight et al., 1998; Sung et al., 2000; Meng et al., 2001; Lee et al., 2003; Gao et al., 2004)
    - Orthographic transliteration (Haizhou et al., 2004)
  - Transliteration model
    - Capture the knowledge of bilingual phonetic association
- Our Approach
  - Orthographic Transliteration
  - Modified Joint Source-Channel Model

Applications of NE Transliteration

- Multilingual NE and term processing
- Machine translation
- Corpus alignment
- Cross lingual information retrieval
- Automatic bilingual dictionary compilation
- Automatic name transliteration

Problems for NERC in Indian Languages

- Lacks capitalization information
- More diverse Indian person names
  - Lot of person names appear in the dictionary with other specific meanings
    - For e.g., *KabiTA* (Person name vs. Common noun with meaning ‘poem’)
- High inflectional nature of Indian languages
  - Richest and most challenging sets of linguistic and statistical features resulting in long and complex wordforms
- Scarcity of Corpus and NE annotated corpus
- Free word order nature of Indian languages
- Resource-constrained environment of Indian languages
  - POS taggers, morphological analyzers, name lists etc. are not available in the web
- Non-availability of sufficient published works

NE Tagset

- Reference Point- CoNLL 2003 shared task tagset
- Tagset: 4 NE tags
  - Person name
  - Location name
  - Organization name
  - Miscellaneous name (date, time, number, percentages, monetary expressions and measurement expressions)

- IJCNLP-08 NERSSEAL Shared Task Tagset: Fine-grained 12 NE tags (available at http://ltrc.iiit.ac.in/ner-ssea-08)

- Tagset Mapping (12 NE tags→4 NE tags)
  - ❑ NEP → Person name
  - ❑ NEL→ Location name
  - ❑ NEO→ Organization name
  - ❑ NEN [number], NEM [Measurement] and NETI [time]→Miscellaneous name
  - ❑ NETO [title-object], NETE [term expression], NED [designations], NEA [abbreviations], NEB [brand names], NETP [title persons

Resources and Tools for NERC in Bengali

- Web-based Corpus
  - Developed from the newspaper archive
- NE annotated Corpus
  - Manual annotation by me
  - Verified by an expert
- Part of Speech (POS) Taggers
  - Hidden Markov Model (HMM), Maximum Entropy (ME), Conditional Random Field (CRF) and Support Vector Machine (SVM)
  - Datasets: Through our participations in two consecutive POS tagging and chunking shared tasks
- Lexicon
  - Created from the news corpus using an unsupervised approach
  - Size: 128K wordforms
  - Root words and their basic POS information, namely noun, verb, adjective, pronoun and indeclinable (preposition, conjunction and interjection)
- Gazetteers
  - Prepared semi-automatically

Web-based Corpus

- Developed from the web-archive of a widely read Bengali newspaper
- Our Corpus Development Procedure
  - Language resource acquisition using a Web Crawler
    - Retrieves web pages in HTML format from the news archive of a leading Bengali newspaper within a range of dates
    - Hierarchical directory structure (year→ month→ day)
  - Language resource creation that includes Hyper Text Markup Language (HTML) file cleaning and code conversion
    - Identify HTML files containing news documents
    - Discard HTML files that do not contribute to text processing activities
    - Bengali texts in the archive are in dynamic fonts
    - Graphemic to Orthographic Coding
    - Three news archive fonts→ ISCII (Indian Standard Code for Information Interchange) code
  - Language resource annotation that involves defining a tagset and subsequent tagging of the news corpus

- Corpus Size: 34 million wordforms
  - Size can be increased dynamically by day after day

News Corpus Tagset

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

➢Tags are not able to recognize the various NEs that appear within the actual news body

News Corpus Statistics

- We collected news data of
  - 5 years (2001-2005)
- Nature of Corpus-Dynamic and size can be incresed everyday

Total number of news documents in the corpus	108, 305
Total number of sentences in the corpus	2, 822, 737
Avgerage number of sentences in a document	27
Total number of wordforms in the corpus	33, 836, 736
Avgerage number of wordforms in a document	313
Total number of distinct wordforms in the corpus	467, 858

NE annotated Corpus

- Automatic NE tagging (tags present in the web pages of the Bengali news corpus)
  - *date→ Miscellaneous name*
  - *location name→ Location name*
  - *reporter name→ Person name*
  - *agency name→ Organization name*

- Limitation: Able to identify NEs that appear in some fixed places

- Manual NE tagging (Part of the Bengali news corpus)
  - Coarse-grained tagset: Four NE tags
    - *Person name*, *Location name*, *Organization name* and *Miscellaneous name*
    - Corpus collected from the Politics, Sports and National domains

  - Fine-grained NE tagset: Twelve NE tags of IJCNLP-08 Shared Task on NER for South and South East Asian Languages (NERSSEAL)

  - Sanchay Editor (available at *sourceforge.net/project/nlp-sanchay sourceforge.net/project/nlp-sanchay*) , a text editor for the Indian languages

Coarse-grained NE Tagged Corpus Statistics (Manual)

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

Statistics of the 200K-tagged Corpus

NE Tag	#Wordforms	#distinct wordforms	Avg. Length of NE
Person name	10,032	6,663	1.87
Location name	4,123	2,129	1
Organization name	1,119	674	2.23
Miscellaneous name	4,475	2,786	1.09

Distribution of the individual NE tags

Gazetteers

- First names: 72,206 entries
- Middle names: 2,491 entries
- Last names: 5,288 entries
- Location names: 8,885 entries
- Organization names: 3,576 entries
- Organization suffix word: 94 entries
- Person prefix word: 145 entries
- Common location: 147 entries
- Designations: 139 entries
- Action verbs: 141 entries
- Common word: 521 entries
- Function words: 743 entries
- Measurement clue words: 52 entries
- Month names :24 entries
- Weekdays : 14 entries
- NE suffixes: 115 entries
- Noun inflections: 27 entries
- Verb inflections: 214 entries
- Adjective inflections: 92 entries

Part of Speech (POS) Tagging in Bengali

- Approaches of POS tagging
  - Hidden Markov Model (HMM)
  - Maximum Entropy (ME)
  - Conditional Random Field (CRF)
  - Support Vector Machine (SVM)

- Datasets and POS Tagset for POS Tagging
  - Natural Language Processing Association of India Machine Learning (NLPAI) Contest 2006 data (http://ltrc.iiit.net/nlpai_contest06): POS tagging and Chunking for Indian languages
  - Shallow Parsing on South and South East Asian Languages (SPSAL) 2007 Contest data (http://shiva.iiit.ac.in/SPSAL2007/contest.php)-POS tagging and Chunking for South and South East Asian languages (Workshop conducted as part of IJCAI-07)
  - POS Tagset: 27 tags (http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf)

Datasets for POS Tagging

- Data Sets:
  - Number of tokens: 72,341
  - Tagset: 27 POS tags, defined for the Indian languages
  - Source of data: Participations in
    - NLPAI ML- 2006 (http://ltrc.iiitnet/nlpai contest06/data2) contest: 46,923 tokens
    - SPSAL-2007 (http://shiva.iiit.ac.in/SPSAL2007) contest: 25,418 tokens

Set	Number of tokens
Training	57,341
Development	15,000
Test	35,000

Training, development and test sets

POS Tagging Experiments

- Same set of features for ME, CRF and SVM

Model	Best Set of Features	Accuracy (in %)
HMM	Second order model, 1st order contextual information to emission probability	84.56
ME	Context window of size three (i.e., previous, current and next words), prefixes and suffixes of length up to three characters of the current word only, POS information of the previous word, NE tag of the current word, Lexicon, Symbol, Function word and digit	87.06
CRF	Context window of size five (i.e., preceding two words, current word and next two words), prefixes and suffixes of length up to three characters of the current word only, POS information of the previous word, NE tags of the current word and previous words, Lexicon, Symbol, Function word and digit	89.84
SVM	Context window of size six (i.e., previous three words, current word and the next two words), prefixes and suffixes of length up to three characters of the current word only, POS information of the previous two words, NE tags of the current and previous words, Lexicon, Symbol, Function word and digit	90.12

Active Learning based NERC System

- Four different models
  - Model A: Lexical context patterns learnt from the unlabeled corpus
  - Model B: Lexical context patterns learnt from the unlabeled corpus + linguistic features
  - Modified Model A and Modified Model B:
    - Assumption: Seed name serves as a
      - *positive example* for its own NE class
      - *negative example* for other NE classes
      - *error example* for non-NEs

Active Learning based NERC System (Contd..)

- Preparation of seed lists for each of the NE tag
  - *reporter-->Person name*, *location→ Location name* and *agency→ Organization name* tags of the Bengali news corpus
- Tagging against seed lists and/or clue words
  - Left and right tags around each occurrence of the seed NEs
  - Model A and Modified Model A: Training corpus tagged only with seed entities
  - Model B and Modified Model B: Training corpus tagged with seed entities + gazetteers + linguistic rules
- Lexical pattern generation from the tagged NEs in the training corpus
  - For each tag T, *lexical* pattern *p* generated using a context window of maximum width 4 (excluding the tagged NE) around the left and the right tags
- Generate further patterns in a bootstrapping manner until no new patterns can be generated
  - Matching of every pattern p of P against the training corpus
  - Various word inflections considered during pattern matching
  - Determination of NE boundary (Heuristics and/or POS information)
  - Manual checking of new NE
  - Apply bootstrapping until no new patterns generated

Evaluation Results (Datasets)

- Training set: Unlabeled 10 million wordforms collected from the Bengali news corpus (Ekbal and Bandyopadhyay, 2008a)
- Test set: Gold standard 35K worforms

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

Training set statistics

β

Evaluation Technique

- Evaluation Parameters:
  - ➢ CoNLL-2003 Shared Task on Language Independent NERC (Tjong Kim Sang and Meulder, 2003)
  - ➢ *Recall*, *Precision* and *F-Score* (or, F_β)

$$Recall = \frac{\text{Number of NEs detected by the system}}{\text{Number of NEs present in the gold standard test set}} \times 100\%$$

$$Precision = \frac{\text{Number of detected NEs that are correct}}{\text{Number of NEs detected by the system}} \times 100\%$$

$$F_\beta = \frac{(\beta^2 + 1) \times Recall \times Precision}{\beta^2 \times Precision + Recall} \times 100\%$$

$$F_{\beta=1} = \frac{(\beta^2 + 1) \times Recall \times Precision}{\beta^2 \times Precision + Recall} \times 100\% = \frac{2 \times Recall \times Precision}{Precision + Recall}$$

- $F_\beta \rightarrow$ weighting between *Recall* and *Precision*
- Class of measures introduced by Van Rijsbergen (1975)
- *F-Score* measure combines *Recall* and *Precision* with an equal weight and hence is the harmonic mean of the two quantities
- β =1

Evaluation Results (Contd..)

- Evaluation Procedure
  - Each pattern of the *Accept Pattern Set* matched against the test set
  - Identified NEs assigned appropriate NE category

Model	Recall (in %)	Precision (in %)	F-Score (in %)
A (Baseline)	64.32	67.29	65.77
B	66.07	69.11	67.56
Modified A	66.19	70.12	68.11
Modified B	68.11	71.37	69.12

Supervised NERC Systems

- DataSets
  - Manually annotated
    - 200K wordforms of the Bengali news corpus with *Person name*, *Location name*, *Organization name* and *Miscellaneous name*
    - *Misecellaneous name*→ date, time, number, monetary expressions and measurement expressions
    - Domain: International, National, State and Sports
    - The annotation carried out by me and verified by a linguistic expert
  - IJCNLP-08 Shared Task on Named Entity Recognition for South and South East Asian Languages (NERSSEAL) (http://ltrc.iiit.ac.in/ner-ssea-08):
    - 122K wordforms tagged with a fine-grained tagset of 12 tags
    - Tagset mapping: 12 NE tags→4 NE tags

Supervised NERC Systems

IJCNLP-08 NERSSEAL Shared Task Tag	Coarse-grained Tag
NEP	*Person name*
NEL	*Location name*
NEO	*Organization name*
NEN, NEM, NETI	*Miscellaneous name*
NEA, NED, NEB, NETP, NETE, NETO	*NNE*

Tagset mapping

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

Training, Development and Test Sets

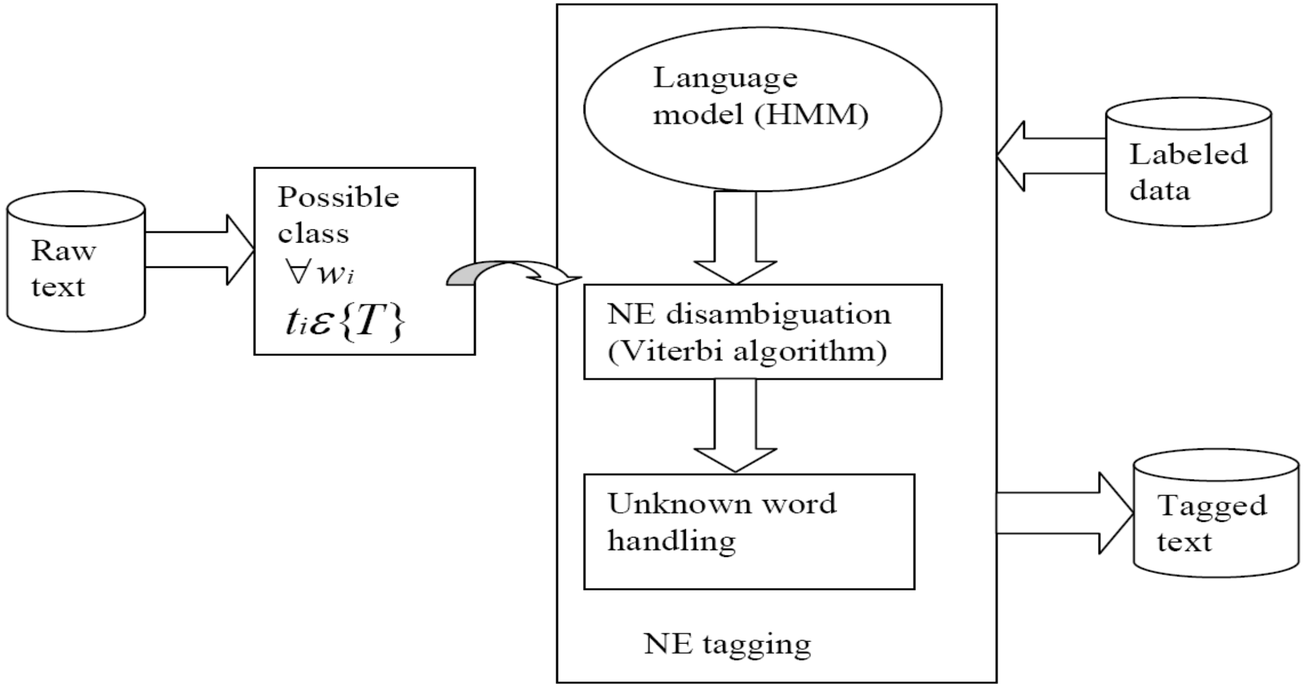
Supervised NERC Systems (B-I-E Format)

NE tag	Meaning	Example
PER	Single-word person name	*sachin* / PER
LOC	Single-word location name	*jadavpur*/LOC
ORG	Single-word organization name	*infosys* / ORG
MISC	Single-word miscellaneous name	100%/ MISC
B-PER I-PER E-PER	Beginning, Internal or the End of a multiword person name	*sachin*/B-PER *ramesh*/I-PER *tendulka* r/ E- PER
B-LOC I-LOC E-LOC	Beginning, Internal or the End of a multi-word location name	*mahatma*/B-LOC *gandhi*/I-LOC *road*/E-LOC
B-ORG I-ORG E-ORG	Beginning, Internal or the End of a multi-word organization name	*bhaba*/B-ORG *atomic*/I-ORG *research*/I-ORG *center*/E-ORG
B-MISC I-MISC E-MISC	Beginning, Internal or the End of a multi-word miscellaneous name	*10e*/B-MISC *magh*/ I-MISC *1402*/E-MISC
NNE	Words that are not named entities (“none-of–the-above” category)	*neta*/NNE, *bidhansabha*/NNE

Hidden Markov Model (HMM) based NERC System

- HMM-
  - Statistical construct used to solve classification problems, having an inherent state sequence representation
  - Transition probability: Probability of traveling between two given states
  - A set of output symbols (also known as *observation*) emitted by the process
  - Emitted symbol depends on the probability distribution of the particular state
  - Output of the HMM: Sequence of output symbols
  - Exact state sequence corresponding to a particular observation sequence is unknown (i.e., *hidden*)
  - Simple language model (*n-gram*) for NE tagging
    - Uses very little amount of knowledge about the language, apart from simple context information

HMM based NERC System (Contd..)



HMM based NERC Architecture

HMM based NERC System (Contd..)

- Components of HMM based NERC system
  - Language model
    - Represented by the model parameters of HMM
    - Model parameters estimated based on the labeled data during supervised learning
  - Possible class module
    - Consists of a list of lexical units associated with the list of 17 tags
  - NE disambiguation algorithm
    - Input: List of lexical units with the associated list of possible tags
    - Output: Output tag for each lexical unit using the encoded information from the language model
    - Decides the best possible tag assignment for every word in a sentence according to the language model
    - Viterbi algorithm (Viterbi, 1967)
  - Unknown word handling
    - Viterbi algorithm (Viterbi, 1967) assigns some tags to unknown words
    - variable length NE suffixes
    - Lexicon (Ekbal and Bandyopadhyay, 2008d)

HMM based NERC System (Contd..)

- Problem of NE tagging

Let W be a sequence of words

$$W = w_1 , w_2 , \ldots , w_n$$

Let T be the corresponding NE tag sequence

$$T = t_1 , t_2 , \ldots , t_n$$

Task : Find T which maximizes $P (T | W)$

$$T' = \operatorname{argmax}_T P (T | W)$$

HMM based NERC System (Contd..)

By Bayes Rule,

$P (T | W) = P (W | T) * P (T) / P (W)$

$T' = \text{argmax}_T \, P (W | T) * P (T)$

- Models
  - Fisrt order model (Bigram): The probability of a tag depends only on the previous tag
  - Second order model (Trigram): The probability of a tag depends on the previous two tags

- Transition Probability

Bigram→ $P (T) = P (t_1) * P (t_2 | t_1) * P (t_3 | t_1 \, t_2) \ldots\ldots * P (t_n | t_1 \ldots t_{n-1})$

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

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

Where, \$→dummy tag used to represent the beginning of a sentence

HMM based NERC System (Contd..)

- Estimation of unigram, bigram and trigram probabilities from the training corpus

Unigram : $P(t_3) = \frac{freq(t_3)}{N}$

Bigram : $P(t_3 \mid t_2) = \frac{freq(t_2, t_3)}{freq(t_2)}$

Trigram : $P(t_3 \mid t_1, t_2) = \frac{freq(t_1, t_2, t_3)}{freq(t_1, t_2)}$

- Emission Probability

$$P(W \mid T) \approx P(w_1 \mid t_1) * P(w_2 \mid t_2) * \ldots * P(w_n \mid t_n)$$

- Estimation : $P(w_i \mid t_i) = \frac{freq(w_i, t_i)}{freq(t_i)}$

HMM based NERC System (Contd..)

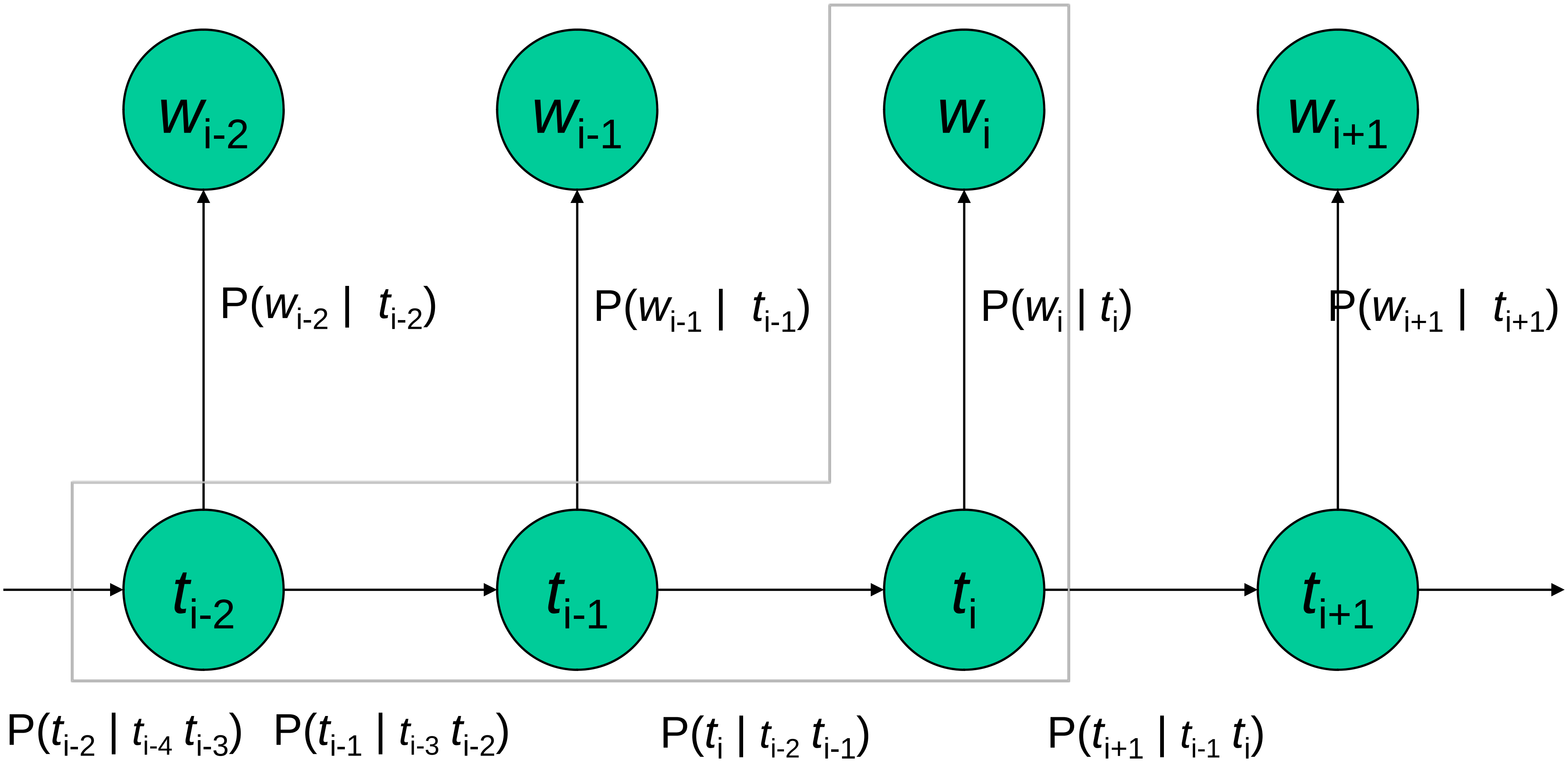
- Context Dependency (Our Modification)

  – To make Markov model powerful, we introduce a 1st order context dependent feature

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

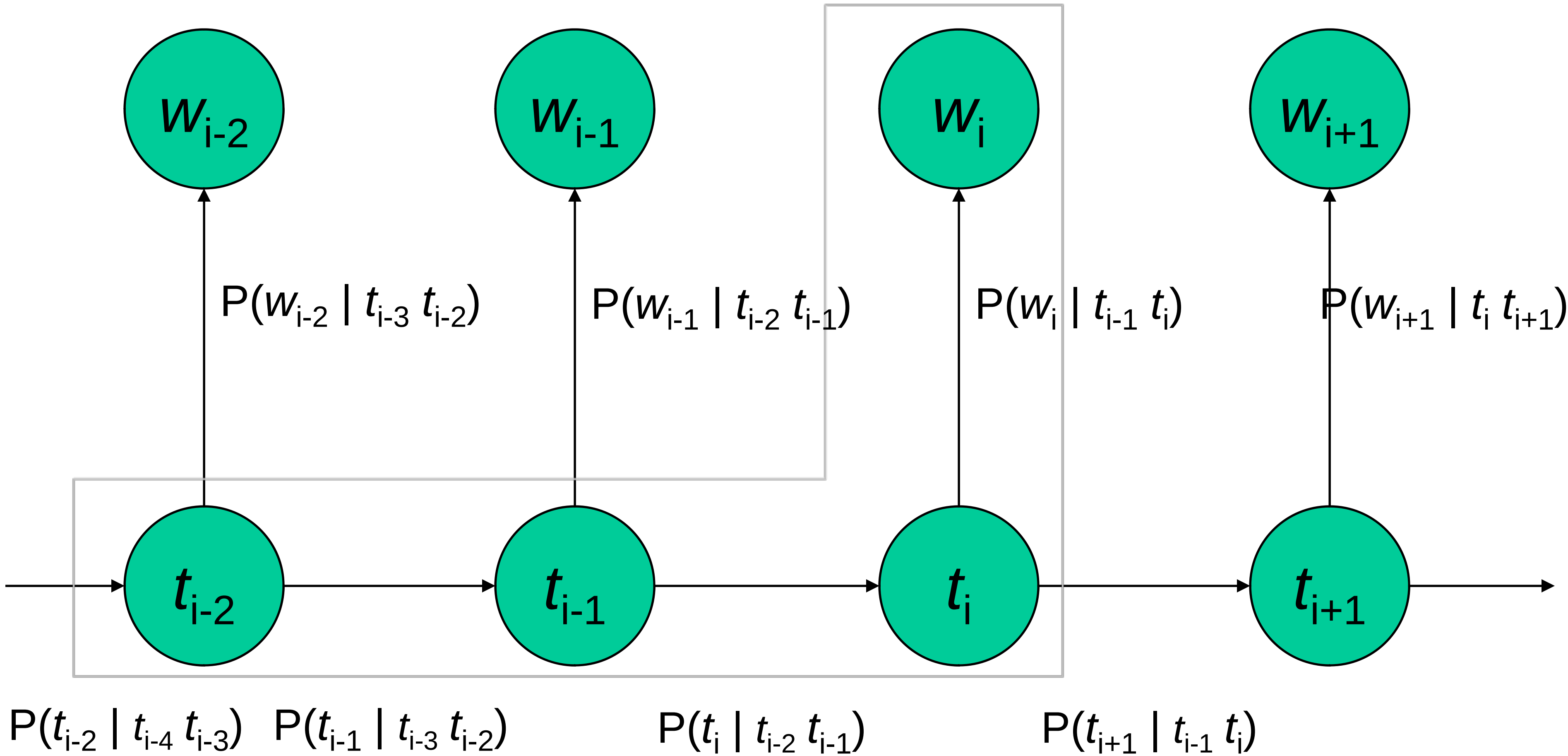
$$P(w_i \mid t_{i-1}, t_i) = \frac{freq(t_{i-1}, t_i, w_i)}{freq(t_{i-1}, t_i)}$$

HMM based NERC System (Contd..)



2nd order Hidden Markov Model

HMM based NERC System (Contd..)



2nd order Hidden Markov Model (Proposed)

HMM based NERC System (Contd..)

- Why Smoothing?
  - All events may not be encountered in the limited training corpus
  - Insufficient instances for each *bigram* or *trigram* to reliably estimate the probability
  - Setting a probability to zero has an undesired effect
- Procedure
  - Transition probability :

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

$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

  - Emission probability :

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

$$\theta_1 + \theta_2 = 1$$

  - Calculation of λs and Ѳs (Brants, 2000)

HMM based NERC System (Contd..)

- Handling of unknown words
  - Viterbi algorithm (Viterbi, 1967) attempts to assign a tag to the unknown words
  - $P(w_i \mid t_i) \rightarrow P(f_i \mid t_i)$
    - Calculated based on the features of unknown word
    - Suffixes: Probability distribution of a particular suffix with respect to specific NE tags is generated from all words in the training set that share the same suffix
      - →Variable length person name suffixes (e.g., - *bAbu*[-babu], -*dA* [-da] , -*di*[-di] etc)
      - →Variable length location name suffixes (e.g., - *lYAnd*[-land], -*pur*[-pur], -*liYA*[-lia]) etc)

→Lexicon

- → 128,000 entries
- → Lexicon contains the root words and their basic POS information such as: noun, verb, adjective, adverb, pronoun and indeclinable (preposition, conjunction and interjection)
- →Unknown word that is found to appear in the lexicon is most likely not a NE

Results of the HMM based System

Model	Reacall (in %)	Precision (in %)	F-Score (in %)
HMM (*bigram*)	76.92	74.79	75.84
HMM (*trigram*)	77.33	75.98	76.65

Results on development set

Observation:

1. Second order model performs better than first order model with a margin of 0.81%

2. Trigram selected to report the test set results

Model	Reacall (in %)	Precision (in %)	F-Score (in %)
Baseline (i.e., Model A)	64.32	67.29	65.77
HMM	77.04	75.17	75.76

Results on the test set

Observation: HMM performs better than the *baseline* model with more than 12.72%, 7.88%, and 9.99% in *Recall*, *Precision*, and *F-Score* values, respectively

Supervised NERC Systems (ME, CRF and SVM)

- Limitations of HMM
  - Use of only local features may not work well
  - Simple HMM models do not work well when large data are not used to estimate the model parameters
  - Incorporating a diverse set features in an HMM based NE tagger is difficult and complicates the smoothing

- Solution:
  - Maximum Entropy (ME) model, Conditional Random Field (CRF) or Support Vector Machine (SVM)
  - ME, CRF or SVM can make use of rich feature information
- ME model
  - Very flexible method of statistical modeling
  - A combination of several features can be easily incorporated
  - Careful feature selection plays a crucial role
  - Does not provide a method for automatic selection of useful features from a given set
  - Features selected using heuristics
  - Adding arbitrary features may result in overfitting

Supervised NERC Systems (ME, CRF and SVM)

- CRF
  - Unlike ME, CRF does not require careful feature selection in order to avoid overfitting
  - Freedom to include arbitrary features
  - Ability of feature induction to automatically construct the most useful feature combinations
  - Conjunction of features (e.g., 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))
  - Infeasible to incorporate all possible conjunction features due to overflow of memory
  - Good to handle different types of data

- SVM
  - Predict the classes depending upon the labeled word examples only
  - Predict the NEs based on feature information of words collected in a predefined window size only
  - Can not handle the NEs outside tokens
  - Achieves high generalization even with training data of a very high dimension
  - Can handle non-linear feature spaces with the use of *kernel function*
  - Good to handle same kind of data

Named Entity Features

- Language Independent Features
  - Can be applied for NERC in any language
- Language Dependent Features
  - Generated from the language specific resources like gazetteers and POS taggers
  - Indian languages are resource-constrained
  - Creation of gazetteers in resource-constrained environment requires *a priori* knowledge of the language
  - POS information depends on some language specific phenomenon such as person, number, tense, gender etc
  - POS tagger (Ekbal and Bandyopadhyay, 2008d) makes use of the several language specific resources such as lexicon, inflection list and a NERC system to improve its performance
- Language dependent features improve system performance

Language Independent Features

- Context Word: Preceding and succeeding words

- Word Suffix
  - Not necessarily linguistic suffixes
  - Fixed length character strings stripped from the endings of words
  - Variable length suffix -binary valued feature
- Word Prefix
  - Fixed length character strings stripped from the beginning of the words

- Named Entity Information: Dynamic NE tag (s) of the previous word (s)

- First Word (binary valued feature): Check whether the current token is the first word in the sentence

Language Independent Features (Contd..)

- Length (binary valued): Check whether the length of the current word less than three or not (shorter words rarely NEs)
- Position (binary valued): Position of the word in the sentence
- Infrequent (binary valued): Infrequent words in the training corpus most probably NEs
- Digit features: Binary-valued
  - Presence and/or the exact number of digits in a token
    - CntDgt : Token contains digits
    - FourDgt: Token consists of four digits
    - TwoDgt: Token consists of two digits
    - CnsDgt: Token consists of digits only

Language Independent Features (Contd..)

- Combination of digits and punctuation symbols
  - CntDgtCma: Token consists of digits and comma
  - CntDgtPrd: Token consists of digits and periods

- Combination of digits and symbols
  - CntDgtSlsh: Token consists of digit and slash
  - CntDgtHph: Token consists of digits and hyphen
  - CntDgtPrctg: Token consists of digits and percentages

- Combination of digit and special symbols
  - CntDgtSpl: Token consists of digit and special symbol such as $, # etc.

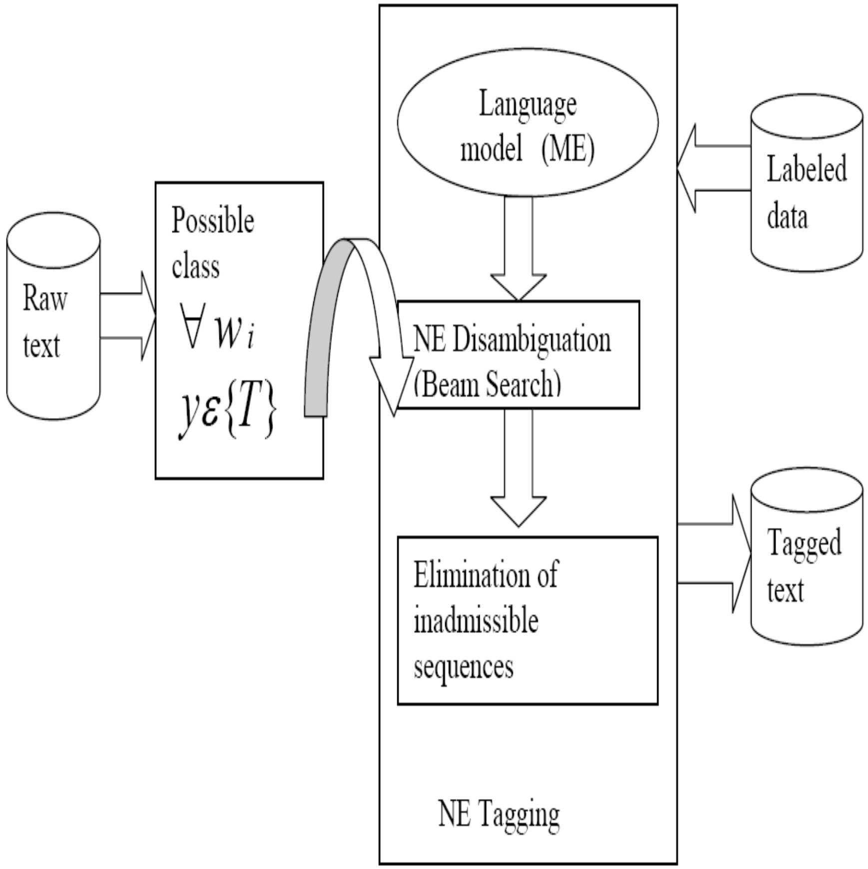
Language dependent Features (Contd..)

- Part of Speech (POS) Information: POS tag(s) of the current and/or the surrounding word(s)

    - SVM-based POS tagger (Ekbal and Bandyopadhyay, 2008b)
    - Accuracy=90.2%
    - SVM based NERC→POS tagger developed with a fine-grained tagset of 27 tags
    - ME and CRF based NERC→ Coarse-grained POS tagger
        - Nominal, PREP (Postpositions) and Other

- Gazetteer based features (binary valued): Several features extracted from the gazetteers

ME based NERC System



- **Language model**: Represented by ME model parameters
- **Possible class module**: Consists of a list of lexical units for each word associated with the list of 17 tags
- **NE disambiguation**: Decides the most probable tag sequence for a given word sequence
  - *beam search* algorithm
- **Elimination of inadmissible sequences**: Removes the inadmissible tag sequences from the output of the ME model

Tool: C++ based ME Package (http://homepages.inf.ed.ac.uk/s0450736/software/maxent/maxent-20061005.tar.bz2)

ME based NERC System (Contd..)

- Elimination of Inadmissible Tag Sequences
  - Inadmissible tag sequence (e.g., B-PER followed by LOC)
  - Transition probability

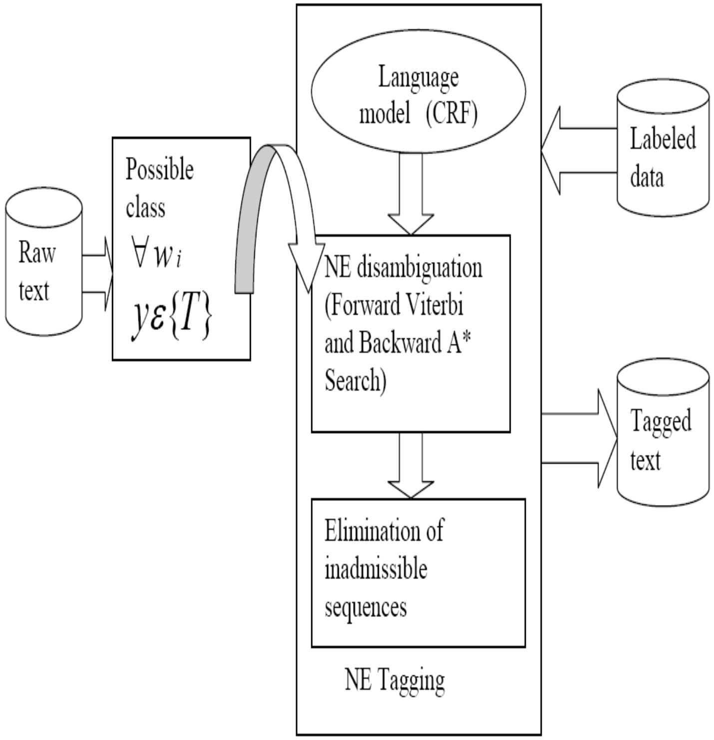
$P(c_i \mid c_j)$ =1, if the sequence is admissible
=0, otherwise

  - Probability of the classes assigned to the words in a sentence 's' in a document 'D' defined as :

$$P(c_1 \ldots, c_n \mid s, D) = \prod_{i=1}^{n} P(c_i \mid s, D) * P(c_i \mid c_{i-1})$$

where, $P(c_i \mid s, D)$ is determined by the maximum entropy classifier

CRF based NERC System (Our Approach)



- Language model: Represented by CRF model parameters
- **Possible class module**: Consists of a list of lexical units for each word associated with the list of 17 tags
- **NE disambiguation**: Decides the most probable tag sequence for a given word sequence
  - Forward Viterbi and backword A* search algorithm (Rabiner, 1989) for disambiguation
- **Elimination of inadmissible sequences**: Removes the inadmissible tag sequences from the output of the CRF model (Same as ME model)

Tool: C^{++} based CRF++ package (http://crfpp.sourceforge.net)

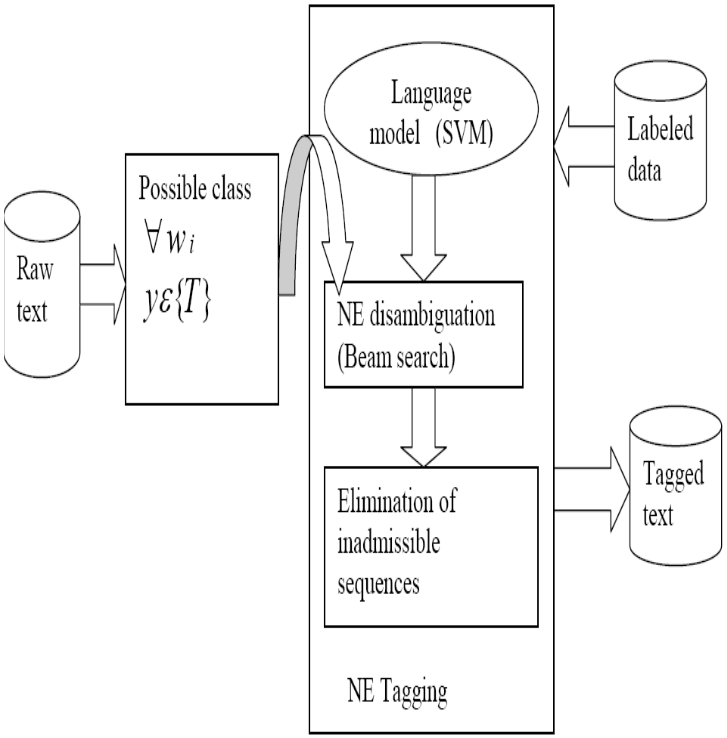
CRF based NERC System (Contd..)

- Feature Template: Feature represented in terms of feature template

$w_{(i-m)}$
$w_{(i-m-1)}$
….
w_{i-1}
w_i
w_{i+1}
…..
$w_{(i+n-1)}$
$w_{(i+n)}$
Combination of w_{i-1} and w_i
Combination of w_i and w_{i+1}
Dynamic output NE tag (t_i) of the previous word
Feature vector of w_i
POS tag(s) of the surrounding word(s)
Gazetteer based features

Feature template used in the experiment

SVM based NERC System (Our Approach)



•Language model: Represented by SVM model parameters

•Possible class module: Considers any of the 17 NE tags to each word

•NE disambiguation: Beam search (Selection of *beam* width (i.e., N) is very important, as larger *beam* width does not always give a significant improvement in performance)

•**Elimination of inadmissible tag sequences:** Same as ME and CRF

➢Training: YamCha toolkit (http://chasen-org/~taku/software/yamcha/)

➢Classification: TinySVM-0.07 (http://cl.aist-nara.ac.jp/~taku-ku/software/TinySVM)

➢*one vs rest* and *pairwise* multi-class decision methods

➢*Polynomial kernel function*

SVM based NERC System (Contd..)

$\rightarrow$ Direction $\rightarrow$

Word: w_{i-2} w_{i-1} w_i w_{i+1} w_{i+1}

POS: p_{i-2} p_{i-1} p_i p_{i+1} p_{i+2}

NE: t_{i-2} t_{i-1} t_i

Feature representation in SVM

- w_i → word appearing at the *ith* position
- p_i → POS feature of w_i,
- t_i → NE label for the i^{th} word
- Reverse parsing direction is possible (from right to left)
- Models of SVM:
  - SVM-F: Parses from left to right
  - SVM-B: Parses from right to left

- Features → Surrounding context, such as words, their lexical features, and the various orthographic word-level features as well as the NE labels

Language Independent Evaluation (ME, CRF and SVM)

(Training: 272K, Development: 50K)

Model	Recall	Precision	F-Score
ME	76.22	72.64	74.67
CRF	78.17	75.81	76.97
SVM-F	**79.14**	**77.26**	**78.19**
SVM-B	79.09	77.15	78.11

- Note:
  - Classifiers trained with the language independent features only
  - SVM-F performs best among all the models

Best Feature Sets for ME, CRF and SVM

Model	Feature
ME	Word, Context (Preceding one and following one word), Prefixes and suffixes of length up to three characters of the current word only, Dynamic NE tag of the previous word, First word of the sentence, Infrequent word, Length of the word, Digit features
CRF	Word, Context (Preceding two and following two words), Prefixes and suffixes of length up to three characters of the current word only, Dynamic NE tag of the previous word, First word of the sentence, Infrequent word, Length of the word, Digit features
SVM-F	Word, Context (Preceding three and following two words), Prefixes and suffixes of length up to three characters of the current word only, Dynamic NE tag of the previous two words, First word of the sentence, Infrequent word, Length of the word, Digit features
SVM-B	Word, Context (Preceding three and following two words), Prefixes and suffixes of length up to three characters of the current word only, Dynamic NE tag of the previous two words, First word of the sentence, Infrequent word, Length of the word, Digit features

- Best Feature set Selection:
  - Training with language independent features and tested with the development set

Language Dependent Evaluation (ME, CRF and SVM)

(Training: 272K, Development: 50K)

Model	Recall	Precision	F-Score
ME	87.02	80.77	83.78
CRF	87.63	84.03	85.79
SVM-F	**87.74**	**85.89**	**86.81**
SVM-B	87.69	85.17	86.72

Language Dependent Evaluation (ME, CRF and SVM)

- Observations:

  - Classifiers trained with best set of language independent as well as language dependent features
  - POS information of the words are very effective
    - Coarse-grained POS tagger (Nominal, PREP and Other) for ME and CRF
    - Fine-grained POS tagger (developed with 27 POS tags) for SVM based Systems
    - Best Performance of ME: POS information of the current word only (an improvement of 2.02% F-Score)
    - Best Performance of CRF: POS information of the current, previous and next words (an improvement of 3.04% F-Score)
    - Best Performance of SVM: POS information of the current, previous and next words (an improvement of 2.37% F-Score in SVM-F and 2.32% in SVM-B)

  - NE suffixes, Organization suffix words, person prefix words, designations and common location words are more effective than other gazetteers

Use of Context Patterns as Features

- Use patterns of the Active Learning based NERC system as the features of ME, CRF, SVM and SVM-B

- Words in the left and/or the right contexts of NEs carry effective information for NE identification

- Feature 'ContextInformation' defined by observing the words in the window [−3, 3] (three words spanning to left and right) of the current word
    - 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

Results using Context Patterns as Features

(Training: 272K, Development: 50K)

Model	Recall (in %)	Precision (in %)	F-Score (in %)
ME	88.22	83.71	85.91
CRF	89.51	85.94	87.69
SVM-F	89.67	86.49	88.05
SVM-B	89.61	86.47	88.01

- Observation: Context features effective to improve the overall performance in each of the models
  - ME: 2.13% F-Score
  - CRF: 1.9% F-Score
  - SVM-F: 1.24% F-Score
  - SVM-B:1.29% F-Score
- Context features significantly improve the Precision value in each of the classifiers

Results of ME based NERC System (Contd..)

Model	Recall (in %)	Precision (in %)	F-Score (in %)
A (*baseline*)	64.32	67.29	65.77
ME (LI)	76.13	75.09	75.61
ME (LI + LD)	85.51	81.83	83.63
ME (LI + LD +CONTXT)	86.04	84.98	85.51

Results on the test set

•LI: NER system with language independent features
•LI+LD: NER system with language independent and dependent features
• LI+LD+CONTXT: NER system with language independent, dependent and context features

Observation:

1. Language specific features improve the system performance by 8.02% F-Score
2. Context features improve the system performance by 1.88% F-Score over language dependent version
3. Context features are very effective to improve Precision value (by 3.15%)

Results of the CRF based NERC System (Contd..)

Model	Recall (in %)	Precision (in %)	F-Score (in %)
A (*baseline*)	64.32	67.29	65.77
CRF (LI)	78.13	74.36	76.62
CRF (LI + LD)	87.53	84.54	86.01
CRF (LI + LD +CONTXT)	87.94	87.12	87.53

Results on the test set

Observation:

1. Language dependent features improve the system performance by 9.39% F-Score
2. Context features improve Precision value by 2.58% over the language dependent system
3. Context features improve overall system performance by 1.88% F-Score

Results of the SVM based NERC System (Contd..)

Model	Recall (in %)	Precision (in %)	F-Score (in %)
A (*baseline*)	64.32	67.29	65.77
SVM-F (LI)	79.93	75.49	77.65
SVM-B (LI)	79.71	75.48	77.54
SVM-F (LI + LD)	88.19	83.94	86.01
SVM-B (LI + LD)	88.11	83.84	85.92
SVM-F (LI + LD +CONTXT)	89.91	85.97	87.89
SVM-B (LI + LD +CONTXT)	89.82	85.93	87.83

Results on test set

➢Observation:

- Language dependent features improve 8.36%, and 8.38% F-Scores in SVM-F and SVM-B, respectively
- Quite similar performance of SVM-F and SVM-B
- Improvement over *baseline*
  - SVM-F: 22.12% F-Score
  - SVM-B: 22.06% F-Score

Post-Processing Techniques

- Error analysis for each classifier: Confusion Matrix
- Post-processing techniques defined to reduce the errors of the classifiers
- Post-processing Technique for ME
  - Post-processing the ME output with 8 heuristics
  - Heuristics identified by looking at the nature of the errors
- Post-processing Technique for CRF
  - Assign the correct tag according to the N-best results for every sentence in the test set
  - Here, N=15 (i.e., 15 labeled sequences for each sentence with the confidence scores considered)
  - Collect NEs from the high confident results and then re-assign the tags for low confident results using this NE list
- Post-processing Technique for SVM
  - Class decomposition technique to reduce the uneven class distribution in the training set

Procedure of Post-processing for CRF

S is the set of sentences in the test set, i.e, $S = \{s_1, s_2, ..., s_n\}$;

R is set of n-best result (n=15) of S, i.e, $R = \{r_1, r_2, ..., r_n\}$, 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 j[th] result in r_i.

Creation of NE Set from the High Confident Tags:

for i = 1 to n

{if (r_{i0} >=0.6) then collect all NEs from r_{i0} and add to the set NESet }.

Replacement:

for i=1 to n

{if (r_{i0} >=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)}}.

Class Decomposition for SVM

- Why class decomposition?
  - To remove uneven class distribution
  - Training set: NEs→22,488 wordforms, Non-NEs→249,512 wordforms
  - Leads to the same situation like the *one vs rest* strategy

- Procedure of class decomposition
  - Split 'NNE' (other than NEs) class into several subclasses effectively
  - Decompose 'NNE' class according to the POS information of the word
    - Given a POS tagset→ *POS*
    - Produce new |*POS*| classes, 'NNE-C'|C$\in$*POS*
    - Tag training corpus with a SVM based POS tagger (Ekbal and Bandyopadhyay, 2008d), developed with POS tagset of 27 tags
    - Number of new subclasses → 27 (e.g., 'NNE-NN' (common noun), 'NNE-VFM' (verb finite main) etc)

Results using Post-processing Techniques

Model	Recall (in %)	Precision (in %)	F-Score (in %)
ME	87.29	86. 81	87.05
CRF	89.19	88.85	89.02
SVM-F	90.23	88.62	89.41
SVM-B	90.05	88.61	89.09

- Observation: Post-processing techniques improve the performance significantly
  - ME: Recall=1.25%, Precision=1.83%, F-Score=2.54%
  - CRF: Recall=1.15%, Precision=1.73%, F-Score=1.49%
  - SVM-F: Recall=0.32%, Precision=2.65%, F-Score=1.52%
  - SVM-B: Recall=0.23%, Precision=2.68%, F-Score=1.26%

Semi-Supervised Model for Unlabeled Data Selection

- Goal of the semi-supervised system
  - Reduce the efforts (time and costs) involved in NE annotated data preparation
  - Incremental training
- Overall procedure
  - Selection of unlabeled 35,143 news documents
  - Documents divided
    - news sources/types (i.e., International, National, State, District, Metro [Kolkata], Politics, Sports, Business etc.) to create segments of manageable size
  - Evaluate contribution of each segment separately
  - Reject segments that are not helpful
  - Apply latest updated best model to each subsequent segment

Semi-Supervised Model for Unlabeled Data Selection (Contd..)

- Steps of Semi-supervised model
  - Appropriate document selection
    - Unlabeled data useful if related to the target problem
  - Appropriate sentence selection
    - Majority Voting among ME, CRF, SVM-F and SVM-B
    - Structure of the sentence (i.e., number of words, length of the words etc.)
    - Content of the sentence (i.e., whether contains NE or not)
- Why appropriate document selection?
  - Acquisition of new names and contexts to provide new evidences
  - Old estimates of the models may be worsened
    - Too many incorrect tags added or,
    - Tags incorrect in the context of training and test data
  - Irrelevant data often degrade rather than improve the classifier's performance

Procedure of Document Selection

- Key word construction from the test set *T*
  - Check whether unlabeled document *d* useful or not for T
  - All words of T not considered
- Procedure of key word construction
  - Test T with the CRF based NERC system
  - Query set Q→ All the name candidates in the top N (=10) best hypotheses for each sentence of the test set T
- Relevant document selection
  - Two necessary conditions
    - Document (*d*) must include at least three (heuristically set) names belonging to the set Q
    - Document (*d*) should contain at least seven (heuristic) names

Sentence Selection

- Why sentence selection?
  - Incorrectly tagged or irrelevant sentences degrade model performance
  - Sentences should provide new information compared to the labeled training data
- Procedure of sentence selection
  - Tag relevant documents with the language dependent ME, CRF, SVM-F and SVM-B based NERC systems
  - Apply majority voting
    - Add a sentence to the training set if the majority of models agree to the same output for at least 80% of the words
  - Discard sentence with fewer than five words
  - Discard sentence that does not include any name

Bootstrapping Procedure for Unlabeled Data Selection

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, . . . ,Cn.
3. Consider 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) Select the NE tag of the SVM-F model if the outputs of all the four models differ.
   (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 get reduced in the majority of the models 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 or differs by a heuristic threshold that is set to 0.005)

Impact of Unlabeled Data Selection (Only Document Selection)

Iteration	Sentences Added	F-Score (in %) of the NERC Models			
		ME	CRF	SVM-F	SVM-B
0	0	87.05	89.02	89.41	89.09
1	111	87.41	89.3	89.74	89.51
2	215	87.6	89.47	89.91	89.87
3	313	88.21	89.65	90.12	90.01
4	399	88.53	89.81	90.23	90.14
5	471	88.67	89.92	90.51	90.44
6	563	88.81	90.42	90.73	90.71
7	622	88.93	90.81	90.98	90.79
8	664	89.04	91.12	91.29	91.21
9	694	89.11	91.19	91.53	91.42
10	713	89.27	91.19	91.68	91.69
11	727	89.34	91.19	91.74	91.83
12	741	89.41	91.19	91.84	91.83
13	752	89.41	91.19	92.01	91.83
14	761	89.41	91.19	92.01	91.83

Observation:

•Post-processed models run on 35,143 news documents

•No. of sentences added to the initial training data: 761

•Order of normalization: CRF→SVM-B→ME→SVM-F

•Improvement:

ME: 2.36% F-Score
CRF: 2.17% F-Score
SVM-F: 2.60% F-Score
SVM-B: 2.74% F-Score

Impact of Unlabeled Data Selection (Document and Sentence Selection)

Model		ME	CRF	SVM-F	SVM-B
		F-Score (in %)	F-Score (in %)	F-Score (in %)	F-Score (in %)
1	Post-processed	87.05	89.02	89.41	89.09
2	(1) + Bootstrapping	88.01	89.84	90.05	90.01
3	(2) + Document Selection	88.97	90.89	91.12	91.02
4	(3) + Sentence Selection	89.41	91.19	92.01	91.83

- Rows 2- 3: Without document selection, even though the training corpus size is increased, the performance of the ME, CRF, SVM-F, and SVM-B models decrease by 0.96%, 1.05%, 1.07%, and 1.01% F-Scores
- Conclusion :
  - 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

Multi-Engine System for NERC in Bengali

- Why multi-engine?
  - To achieve better performance
  - A large number of words, tagged wrongly by any model, may be correctly tagged by another model
  - Determine final NE tag from the various models

- Approach for multi-engine
  - Weighted voting
  - Combine ME, CRF, SVM-F and SVM-B
    - Yielded similar performance
    - Determination of appropriate weight for each model

Multi-Engine System for NERC in Bengali (Contd..)

- Weighted Voting techniques
  - Majority voting
    - Same weight assigned to each model
    - Select the classification proposed by the majority of the models
    - Select output of the SVM-F model in case of ties
  - Cross Validation F-Score Values: Assign weight based on the 10-fold cross validation results
    - Total F-Score: Overall average F-Score of any classifier
    - Tag F-Score : Average F-Score value of the individual NE tags as the weight

Results of the Multi-engine System

Voting	Recall (in %)	Precision (in %)	F-Score (in %)
Majority	93.4	92.41	92.9
Total F-Score	93.9	92.91	93.4
Tag F-Score	94.7	92.93	93.8

Overall results

Observations:

•Improvement of 4.39% and 1.79% over ME and SVM-F, respectively

•Improvement of 28.03% over unsupervised *baseline* Model A

NE Category	Recall (in %)	Precision (in %)	F-Score (in %)
Person name	95.61	92.15	93.85
Location name	92.55	89.01	90.75
Organization name	90.12	88.53	89.32
Miscellaneous name	97.19	94.09	95.62

Results of the individual NE tags

Experiments with other Indian Languages

- NER systems in other Indian Languages
  - Hindi, Telugu, Oriya and Urdu

- Approaches
  - HMM, ME, CRF and SVM
  - Language independent features for all the languages
  - Language dependent features for Hindi and Telugu

- Datasets: IJCNLP-08 NERSSEAL Shared Task Data

- Tagset Mapping: 12 NE tags→4 NE Tags

Experiments with other Indian Languages (Contd..)

- Hindi (Training=452,974, Test=32,796)
- Telugu (Training= 54,026, Test= 8,006)
- Oriya (Training= 78,173, Test= 27,007)
- Urdu (Training= 35,447, Test= 12,805)

Language	HMM	ME	CRF	SVM-F	SVM-B
Hindi	73.72	76.71	78.68	79.04	78.86
Telugu	69.04	72.66	74.49	75.94	75.86
Oriya	66.22	68.12	69.65	70.98	70.77
Urdu	61.88	64.24	66.14	65.65	67.15

Conclusion

- Presented an appropriate approach for NERC in Bengali
- Simply supervised machine learning algorithm may not be sufficient to achieve reasonable performance for NERC
- Context patterns generated from the Active Learning Technique effective to improve the performance of the supervised classifiers
- Post-processing the outputs of the classifiers is effective to improve the performance
- Relevant unlabeled data selection is important
- Combination of several classifiers can perform better compared to any single classifier
- Language dependent features improve the system performance
- Semi-supervised model is more suitable for a resource-constrained language

Future Works

- Search for an appropriate clustering algorithm for NERC in resource-constrained languages
- Developing a rule based component to correct the errors of the machine learning based method
- Feature reduction by using the cluster of words as the features in the ME, CRF or SVM models instead of using the words as the features
- Investigation of other effective voting methods
- Use of available 34 million wordfroms for effective document and sentence selection
- Tuning the NER systems for integration into Web People Search, Event Extraction, Emotion Analysis, Sentiment Analysis etc.

Related Publications (Most Relevant)

- Related Publications (Journal)
  - A. Ekbal and S. Bandyopadhyay. 2008. A Web-based Bengali News Corpus for Named Entity Recognition”. In ***Language Resources and Evaluation (LRE) Journal***, Volume 42(2), PP. 173-182, Springer
  - A. Ekbal, S. Naskar and S. Bandyopadhyay (2007). Named Entity Recognition and Transliteration in Bengali. In ***Named Entities: Recognition, Classification and Use, Special Issue of Lingvisticae Investigationes Journal***, 30:1 (2007), 95-114, John Benjamins Publishing Company
  - A. Ekbal and S. Bandyopadhyay (2008). Support Vector Machine Approach for Named Entity Recognition: A Language Independent Approach. In **International Journal of Computer Systems Science and Engineering**, 4(2), pp.155-170
  - A. Ekbal and S. Bandyopadhyay (2008). Maximum Entropy Approach for Named Entity Recognition in Indian Languages. In ***International Journal for Computer Processing on Oriental Languages*** **(IJCPOL)**, Volume 21(3), PP. 1-33, World Scientific Publishing Company
  - A. Ekbal and S. Bandyopadhyay (2009). Named Entity Recognition in Bengali and Hindi using Support Vector Machine. In ***Lingvisticae Investigationes Journal***, John Benjamins Publishing Company (Accepted).

Related Publications (Most Relevant)

- Related Publications (Journal)

  - A. Ekbal and S. Bandyopadhyay (2008). Named Entity Recognition using Appropriate Unlabeled Data, Post-processing and Voting. In **Informatica (Journal of Computing and Informatics)**, ACTA Press (Accepted)

  - **A. Ekbal** and S. Bandyopadhyay (2009). A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi. **Linguistic Issues in Language Technology (LiLT)**, **CSLI Publication**, **Stanford University** (Accepted)

  - **A. Ekbal**, R. Haque and S. Bandyopadhyay (2008). Maximum Entropy Based Bengali Part of Speech Tagging. In A. Gelbukh *(Ed.), Advances in Natural Language Processing and Applications*, *Research in Computing Science (RCS) Journal,* Vol. (33), PP. 67-78

  - **A. Ekbal** and S. Bandyopadhyay (2008). Web-based Bengali News Corpus for Lexicon Development and POS Tagging. In *POLIBITS, an International Journal, ISSN 1870-9044,* Vol. 37(2008), PP. 20-29, National Polytechnic Institute, Mexico

Related Publications (Most Relevant)

- Related Publications (Journal)
  - A. Ekbal, S. Naskar and S. Bandyopadhyay (2007). Named Entity Transliteration. International Journal of **Computer Processing of Oriental Languages** (**IJCPOL**), Vol. 20(4), 289-310, World Scientific Press, Singapore.
  - **A. Ekbal**, S. Naskar and S. Bandyopadhyay (2007). A Generalized Named Entity Transliteration System: Bengali and English as Case Study. In *International Journal of Translation*, Vol. 19, No. 1, Jan-June 2007, PP. 117-132.

- Related Publications (**Conferences and/or Workshops**)
  - **A. Ekbal** and S. Bandyopadhyay (2009). Voted NER System using Appropriate Unlabeled Data. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration* (*NEWS 2009*), ***ACL-IJCNLP 2009***, August 7th, 2009, Singapore, PP. 202-210
  - **A. Ekbal** and S. Bandyopadhyay (2009). Improving the Performance of a NER System by Post-processing, Context Patterns and Voting. In *W. Li and D. Molla-Aliod (Eds): ICCPOL 2009, Lecture Notes in Artificial Intelligence (LNAI), Springer Berlin/Heidelberg, Vol. 5459*, 45-56. (Selected to be published in IJCPOL, WSP)

Related Publications (Most Relevant)

- Publications (Conferences and/or Workshops)
  - **A. Ekbal** and S. Bandyopadhyay (2008). Multi-Engine Approach for Named Entity Recognition in Bengali. In *Proceedings of the 22nd Pacific Asia Conference on Language, Information and Computation (**PACILIC 2008**),* 169-178, Philipines
  - **A. Ekbal** and S. Bandyopadhyay (2008). Appropriate Unlabeled Data, Post-processing and Voting can Improve the Performance of NER System. In *Proceedings of the 6th International Conference on Natural Language Processing (**ICON-08**)*, pp. 234–239, India.
  - **A. Ekbal**, and S. Bandyopadhyay (2008). Bengali Named Entity Recognition using Support Vector Machine. In *Proceedings of the Workshop on Named Entity Recognition for South and South East Asian Languages, **3rd International Joint Conference on Natural Language Processing (IJCNLP-08)***, India, PP: 51-58.
  - **A. Ekbal**, R. Haque and S. Bandyopadhyay (2008). Named Entity Recognition in Bengali: A Conditional Random Field Approach. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (**IJCNLP-08**)*, India, PP: 589-594.

Related Publications (Most Relevant)

- Publications (Conferences and/or Workshops)

  - **A. Ekbal** and S. Bandyopadhyay (2007). Recognition and Transliteration of Bengali Named Entities: A Computational Approach. In *Proceedings of the Recent Advances in Natural Language Processing (**RANLP-2007**)*, 27-29 September, Bulgaria, PP: 1-5.
  - **A. Ekbal**, and S. Bandyopadhyay (2007). Maximum Entropy Approach for Named Entity Recognition in Bengali. In *Proceedings of the 7th International Symposium on Natural Language Processing (**SNLP-07**)*, 1-6, Thailand
  - **A. Ekbal** and S. Bandyopadhyay (2007) Lexical Pattern Learning from Corpus Data for Named Entity Recognition. In the *Proceedings of 5th International Conference on Natural Language Processing* (***ICON 2007***), 4-6 January, Hyderabad, India, PP.123-128
  - A.Ekbal and S. Bandyopadhyay (2008). Development of Bengali Named Entity Tagged Corpus and its use in NER Systems. In *Proceedings of the 6th Workshop on Asian Language Resources, 3rd International Joint Conference on Natural Language Processing (**IJCNLP-08**)*, India, PP: 1-8

Related Publications (Most Relevant)

- Publications (Conferences and/or Workshops)
  - **A. Ekbal** , M. Hasanuzzaman and S. Bandyopadhyay (2009). Voted Approach for Part of Speech Tagging in Bengali. In Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation (***PACLIC-09***), December 3-5, Hong Kong (Accepted)
  - **A. Ekbal** and S. Bandyopadhyay (2008). Part of Speech Tagging in Bengali using Support Vector Machine. In *Proceedings of the International Conference on Information Technology (**ICIT 2008**),* 106-111, IEEE CS Press
  - **A. Ekbal**, R. Haque and S. Bandyopadhyay (2007). Bengali Part of Speech Tagging using Conditional Random Field. In *Proceedings of the 7th International Symposium on Natural Language Processing (**SNLP-07**)*, 131-136, Thailand
  - **A. Ekbal**, S. Mondal and S. Bandyopadhyay (2007). POS Tagging using HMM and Rule-based Chunking. In Proceedings of the Workshop on Shallow Parsing in South Asian Languages, International Joint Conference on Artificial Intelligence (**IJCAI 2007**), 6-12 January 2007, Hyderabad, India, PP. 25-28