Named Entity Recognition and Classification

Asif Ekbal

AI-NLP-ML Group
Dept. of Computer Science and Engineering
IIT Patna, India-800 013
Email: asif@iitp.ac.in
asif.ekbal@gmail.com

Outline

- Background
- Introduction to the various issues of NER
- NER in different languages
- NER in Indian languages
- NER in Specific Domains: Few Examples

- Weighted Vote based Classifier Ensemble
  - Introduction to GA
  - Some Issues of Classifier Ensemble

Outline

- NER in Biomedicine
  - Introduction
  - NE Extraction in Biomedicine (*Weighted Voted Ensemble!*)
  - Issues in Corpus Compatibilities
  - Stacked Ensemble

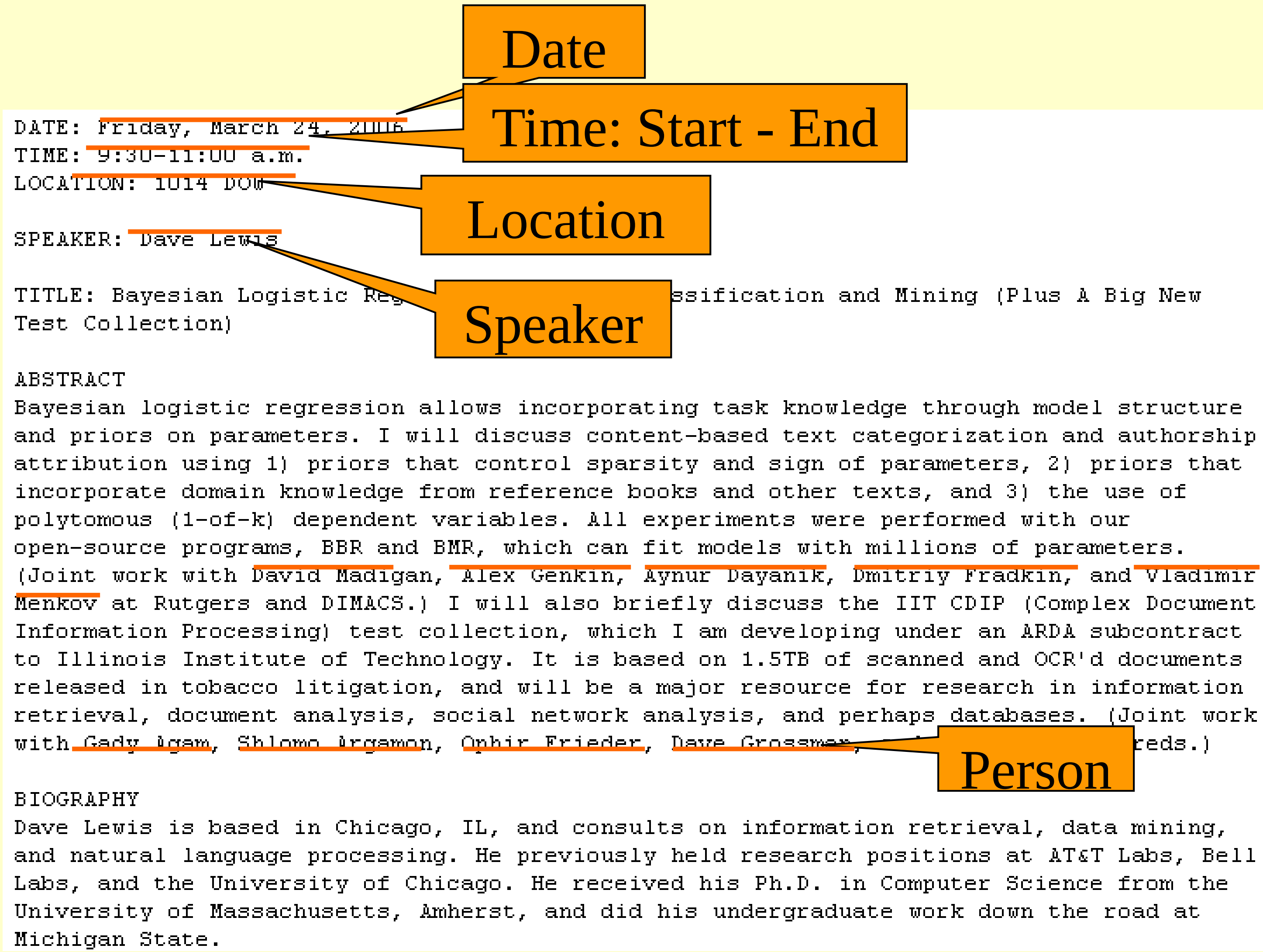
Background

Background: Information Extraction

- To extract information that fits pre-defined database schemas or templates, specifying the output formats

- **IE Definition**
  - **Entity**: an object of interest such as a person or organization
  - **Attribute**: A property of an entity such as name, alias, descriptor or type
  - **Fact**: A relationship held between two or more entities such as Position of Person in Company
  - **Event**: An activity involving several entities such as terrorist act, airline crash, product information

The Problem



DATE: Friday, March 24, 2006
TIME: 9:30-11:00 a.m.
LOCATION: 1014 DOW

SPEAKER: Dave Lewis

TITLE: Bayesian Logistic Reg… ssification and Mining (Plus A Big New Test Collection)

ABSTRACT
Bayesian logistic regression allows incorporating task knowledge through model structure and priors on parameters. I will discuss content-based text categorization and authorship attribution using 1) priors that control sparsity and sign of parameters, 2) priors that incorporate domain knowledge from reference books and other texts, and 3) the use of polytomous (1-of-k) dependent variables. All experiments were performed with our open-source programs, BBR and BMR, which can fit models with millions of parameters. (Joint work with David Madigan, Alex Genkin, Aynur Dayanik, Dmitriy Fradkin, and Vladimir Menkov at Rutgers and DIMACS.) I will also briefly discuss the IIT CDIP (Complex Document Information Processing) test collection, which I am developing under an ARDA subcontract to Illinois Institute of Technology. It is based on 1.5TB of scanned and OCR'd documents released in tobacco litigation, and will be a major resource for research in information retrieval, document analysis, social network analysis, and perhaps databases. (Joint work with Gady Agam, Shlomo Argamon, Ophir Frieder, Dave Grossman, … reds.)

BIOGRAPHY
Dave Lewis is based in Chicago, IL, and consults on information retrieval, data mining, and natural language processing. He previously held research positions at AT&T Labs, Bell Labs, and the University of Chicago. He received his Ph.D. in Computer Science from the University of Massachusetts, Amherst, and did his undergraduate work down the road at Michigan State.

What is "Information Extraction"?

As a task: **Filling slots in a database from sub-segments of text.**

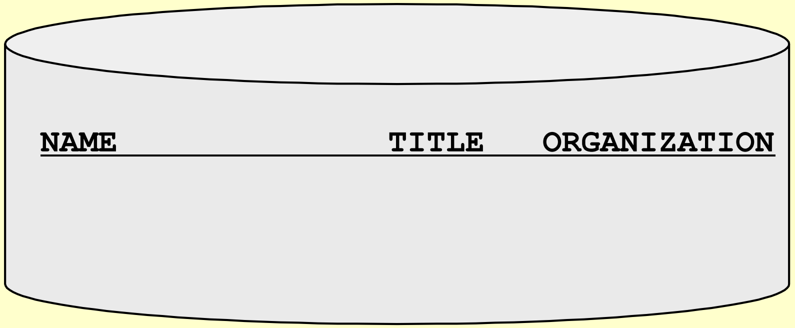
October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…



Courtesy of William W. Cohen

What is "Information Extraction"?

As a task: **Filling slots in a database from sub-segments of text.**

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

IE

NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..

Courtesy of William W. Cohen

What is "Information Extraction"?

Information Extraction =
segmentation + classification + association + clustering

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

Microsoft Corporation
CEO
Bill Gates
Microsoft
Gates
Microsoft
Bill Veghte
Microsoft
VP
Richard Stallman
founder
Free Software Foundation

aka "named entity extraction"

Courtesy of William W. Cohen

What is "Information Extraction"?

Information Extraction =
segmentation + classification + association + clustering

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

Microsoft Corporation
CEO
Bill Gates
Microsoft
Gates
Microsoft
Bill Veghte
Microsoft
VP
Richard Stallman
founder
Free Software Foundation

Courtesy of William W. Cohen

What is "Information Extraction"?

Information Extraction =
segmentation + classification + association + clustering

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…

Microsoft Corporation **CEO** **Bill Gates**
Microsoft **Gates**
Microsoft **Bill Veghte** **Microsoft** **VP**
Richard Stallman **founder** **Free Software Foundation**

Courtesy of William W. Cohen

What is "Information Extraction"?

Information Extraction =
segmentation + classification + association + clustering

October 14, 2002, 4:00 a.m. PT

For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access.“

Richard Stallman, founder of the Free Software Foundation, countered saying…

* **Microsoft Corporation**
  CEO
  Bill Gates
* **Microsoft**
  Gates
* **Microsoft**
  Bill Veghte
* **Microsoft**
  VP

Richard Stallman
founder
Free Software Foundation

NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..

Courtesy of William W. Cohen

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)

Named Entity Recognition

Markables (as defined in MUC6 and MUC7)

- Names of **organization**, **person**, **location**
- Mentions of **date** and **time**, **money** and **percentage**

Example:

"Ms. **Washington**'s candidacy is being championed by several powerful lawmakers including her boss, Chairman **John Dingell** (D., **Mich**.) of the **House Energy and Commerce Committee**."

Task Definition

- Other common types: measures (percent, money, weight etc), email addresses, web addresses, street addresses, etc.
- Some domain-specific entities: names of drugs, medical conditions, names of ships, bibliographic references etc.
- MUC-7 entity definition guidelines (Chinchor'97)

http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ne_task.html

Basic Problems in NER

- Generative in nature
- Variation of NEs – e.g. Prof Manning, Chris Manning, Dr Chris Manning
- Ambiguity of NE types:
  - Washington (location vs. person)
  - May (person vs. month)
  - Ford (person vs organization)
  - 1945 (date vs. time)
- Ambiguity with common words, e.g. "*Kabita*"

  \- Name of person vs. poem

More complex problems in NER

- Issues of style, structure, domain, genre etc.
- Punctuation, spelling, spacing, formatting, … all have an impact:

Dept. of Computing and Maths

Manchester Metropolitan University

Manchester

United Kingdom

Applications

- Intelligent document access
  - Browse document collections by the entities that occur in them
  - Application domains:
    - News
    - Scientific articles, e.g, MEDLINE abstracts
- Information retrieval and extraction
  - Augmenting a query given to a retrieval system with NE information, more refined information extraction is possible
  - For example, if a person wants to search for document containing '*kabiTA*' as a proper noun, adding the NE information will eliminate irrelevant documents with only '*kabiTA*' as a common noun

Applications

- Machine translation
  - NER plays an important role in translating documents from one language to other
  - Often the NEs are transliterated rather than translated
  - For example, '*yAdabpur bishvabidyAlaYa*' → '*Jadavpur University*'
- Automatic Summarization
  - NEs given more priorities in deciding the summary of a text
  - Paragraphs containing more NEs are most likely to be included into the summary

Applications

- Question-Answering Systems
  – NEs are important to retrieve the answers of particular questions

- Speech Related Tasks
  – In Text to Speech (TTS), NER is important for identifying the number format, telephone number and date format
  – In speech rhythm- necessary to provide a short break after the name of person
  – Solving **Out Of Vocabulary (OOV)** words is important in speech recognition

Corpora, Annotation

Some NE Annotated Corpora

- MUC-6 and MUC-7 corpora - English
- CONLL shared task corpora
  - http://cnts.uia.ac.be/conll2003/ner/ : NEs in English and German
  - http://cnts.uia.ac.be/conll2002/ner/ : NEs in Spanish and Dutch
- ACE – English - http://www.ldc.upenn.edu/Projects/ACE/
- TIDES surprise language exercise (NEs in Hindi)
- NERSSEAL shared task- NEs in Bengali, Hindi, Telugu, Oriya and Urdu (http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5)

Corpora, Annotation

- Biomedical, Biochemical and Health Corpora
  - BioNLP-04 shared task
  - BioCreative shared tasks
  - AiMed
  - I2B2
- NER in Tweet
  - ACL-IJCNLP Workshop on Noisy User-generated Text (W-NUT)

The MUC-7 Corpus

<ENAMEX TYPE="LOCATION">CAPE CANAVERAL</ENAMEX>, <ENAMEX TYPE="LOCATION">Fla.</ENAMEX> &MD; Working in chilly temperatures <TIMEX TYPE="DATE">Wednesday</TIMEX> <TIMEX TYPE="TIME">night</TIMEX>, <ENAMEX TYPE="ORGANIZATION">NASA</ENAMEX> ground crews readied the space shuttle Endeavour for launch on a Japanese satellite retrieval mission.

<p>

Endeavour, with an international crew of six, was set to blast off from the <ENAMEX TYPE="ORGANIZATION|LOCATION">Kennedy Space Center</ENAMEX> on <TIMEX TYPE="DATE">Thursday</TIMEX> at <TIMEX TYPE="TIME">4:18 a.m. EST</TIMEX>, the start of a 49-minute launching period. The <TIMEX TYPE="DATE">nine day</TIMEX> shuttle flight was to be the 12th launched in darkness.

Performance Evaluation

- Evaluation metric – mathematically defines how to measure the system's performance against a human-annotated, gold standard

- Scoring program–implements the metric and provides performance measures
  - For each document and over the entire corpus
  - For each type of NE

The Evaluation Metric

Precision = correct answers/answers produced

Recall = correct answers/total possible correct answers

Trade-off between precision and recall

F-Measure = $(\beta^2 + 1)PR / \beta^2 R + P$

β reflects the weighting between precision and recall, typically $\beta=1$

The Evaluation Metric (2)

Precision =

$$\frac{\text{Correct} + \frac{1}{2}\ \text{Partially correct}}{\text{Correct} + \text{Incorrect} + \text{Partial}}$$

Recall =

$$\frac{\text{Correct} + \frac{1}{2}\ \text{Partially correct}}{\text{Correct} + \text{Missing} + \text{Partial}}$$

NE boundaries are often misplaced, so some partially correct results

Named Entity Recognition

- Handcrafted systems
  - Knowledge (rule) based
    - Patterns
    - Gazetteers
- Automatic systems
  - Statistical
  - Machine learning-*Supervised*, *Semi-supervised*, *Unsupervised*
- Hybrid systems

Pre-processing for NER

- Format detection
- Word segmentation (for languages like Chinese)
- Tokenisation
- Sentence splitting
- Part-of-Speech (PoS) tagging

Comparisons between two Approaches

Knowledge Engineering

- rule based
- developed by experienced language engineers
- makes use of human intuition
- requires only small amount of training data
- development could be very time consuming
- some changes may be hard to accommodate

Learning Systems

- use statistics or other machine learning
- developers do not need LE expertise
- requires large amounts of annotated training data
- annotators are cheap (but you get what you pay for!)
- easily trainable and adaptable to new domains and languages

List lookup approach-baseline

- System that recognises only entities stored in its lists (gazetteers)

- Advantages - Simple, fast, language independent, easy to retarget (just create lists)

- Disadvantages - collection and maintenance of lists, cannot deal with name variants, cannot resolve ambiguity

Shallow Parsing Approach (internal structure)

- Internal evidence–names often have internal structure. These components can be either stored or guessed,

e.g. location:

  - Cap. Word + {City, Forest, Centre, River}

    e.g. Sundarban Forest

  - Cap. Word +{Street, Boulevard, Avenue, Crescent, Road}

    e.g. MG Road

e.g. Person

  - Word + {Kumar, Chandra} + Word

    e.g. Deepak Kumar Gupta

Problems with the shallow parsing approach

- Ambiguously capitalized words (first word in sentence)
  [All American Bank] vs. All [State Police]
- Semantic ambiguity
  Bangalore ek badzA shaher heI (Bangalore is a big city)-Location
  Bangalore shikshak heI (Bangalore is a teacher)-Person
- Structural ambiguity
  [Cable and Wireless] vs. [Microsoft] and [Dell]
  [Center for Computational Linguistics] vs. message from
  [City Hospital] for [John Smith]

Shallow Parsing Approach with Context

- Use of context-based patterns is helpful in ambiguous cases

- “Ratan Tata " and “Tata Sons" are indistinguishable

- But with the phrase “Ratan Tata of Tata Sons" and the Person entity “Ratan Tata" recognised, we can use the pattern "*[Person] of [Organization]*" to identify “Tata Sons“ correctly

Examples of context patterns

- [PERSON] earns [MONEY]
- [PERSON] joined [ORGANIZATION]
- [PERSON] left [ORGANIZATION]
- [PERSON] joined [ORGANIZATION] as [JOBTITLE]
- [ORGANIZATION]'s [JOBTITLE] [PERSON]
- [ORGANIZATION] [JOBTITLE] [PERSON]
- the [ORGANIZATION] [JOBTITLE]
- part of the [ORGANIZATION]
- [ORGANIZATION] headquarters in [LOCATION]
- price of [ORGANIZATION]
- sale of [ORGANIZATION]
- investors in [ORGANIZATION]
- [ORGANIZATION] is worth [MONEY]
- [JOBTITLE] [PERSON]
- [PERSON], [JOBTITLE]

Gazetteer lists for rule-based NER

- Needed to store the indicator strings for the internal structure and context rules
- Internal location indicators – e.g., {*river, mountain, forest*} for natural locations; {*street, road, crescent, place, square*, …} for address locations
- Internal organisation indicators–e.g., company designators {*GmbH, Ltd, Inc*, …}
- Produces Lookup results of the given kind

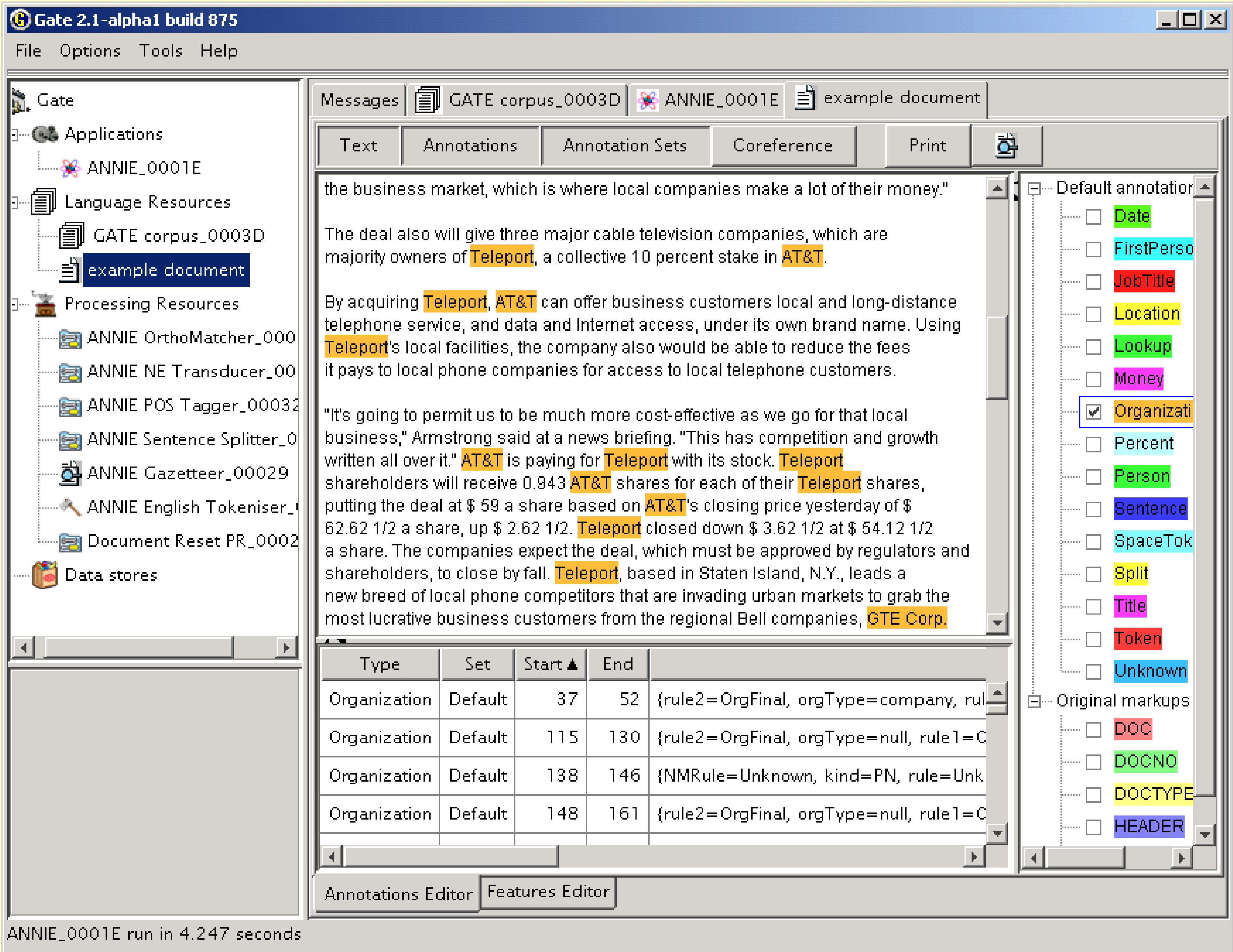
Named Entity Recognition

- Handcrafted systems
  - LTG (Mikheev et al., 1997)
    - F-measure of 93.39 in MUC-7 (the best)
    - Ltquery, XML internal representation
    - Tokenizer, POS-tagger, SGML transducer
  - Nominator (1997)
    - IBM
    - Heavy heuristics
    - Cross-document co-reference resolution
    - Used later in IBM Intelligent Miner

Named Entity Recognition

- Handcrafted systems
  - LaSIE (Large Scale Information Extraction)
    - MUC-6 (LaSIE II in MUC-7)
    - Univ. of Sheffield's GATE architecture (General Architecture for Text Engineering)
  - FACILE (1998)- Fast and Accurate Categorisation of Information by Language Engineering
    - NEA language (Named Entity Analysis)
    - Context-sensitive rules
  - NetOwl (MUC-7)
    - Commercial product
    - C++ engine, extraction rules

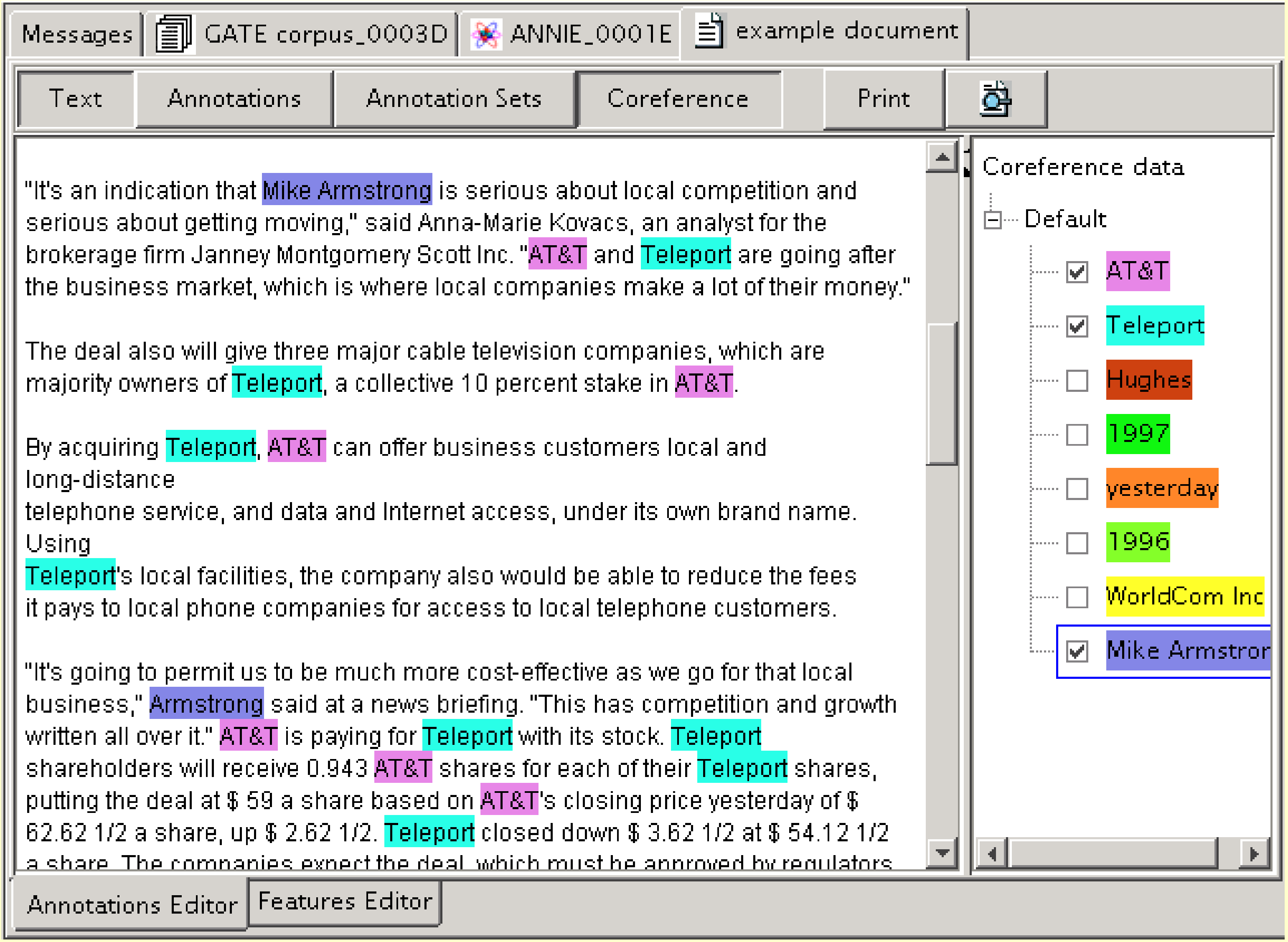
Named Entities in GATE



Using co-reference to classify ambiguous NEs

- Orthographic co-reference module that matches proper names in a document
- Improves NE results by assigning entity type to previously unclassified names, based on relations with classified NEs
- May not reclassify already classified entities
- Classification of unknown entities is very useful for surnames which match a full name, or abbreviations, e.g. [*Tata*] will match [*Sir Jamsedhji Tata*]; [*International Business Machines Ltd.*] will match [*IBM*]

Named Entity Coreference



NER–automatic approaches

- Learning of statistical models or symbolic rules
  - Use of annotated text corpus
    - Manually annotated
    - Automatically annotated

- ML approaches frequently break down the NE task in two parts:
  - Recognising the entity boundaries
  - Classifying the entities in the NE categories

NER – automatic approaches

- Tokens in text are often coded with the IOB scheme
  - O – outside, B-XXX – first word in NE, I-XXX – all other words in NE

  e.g.

India	B-LOC
played	O
with	O
Vivian	B-PER
Richards	I-PER

  - Probabilities:
    - Simple:
      - P(tag i | token i)
    - With external evidence:
      - P(tag i | token i-1, token i, token i+1)

NER–automatic approaches

- Decision trees
  - Tree-oriented sequence of tests in every word
    - Determine probabilities of having a IOB tag
  - Use training data
  - Viterbi, ID3, C4.5 algorithms
    - Select most probable tag sequence
  - SEKINE et al (1998)
  - BALUJA et al (1999)
    - F-measure: 90%

NER – automatic approaches

- HMM-*Generative model*
  - Markov models, Viterbi
  - Works well when large amount of data is available: Nymble (1997) / IdentiFinder (1999)

- Maximum Entropy (ME)-*Discriminative model*
  - Separate, independent probabilities for every evidence (external and internal features) are merged multiplicatively
  - MENE (NYU-1998)
    - Capitalization, many lexical features, type of text
    - F-Measure: 89%

ML features

- The choice of features
  - Lexical features (words)
  - Part-of-speech
  - Orthographic information
  - Affixes (prefix and suffix of any word)
  - Gazetteers

- External, unmarked data is useful to derive gazetteers and for extracting training instances

IdentiFinder [Bikel et al 99]

- Based on Hidden Markov Models
- 7 regions of HMM–one for each *MUC type*, *not-name*, *begin-sentence* and *end-sentence*

- Features
  - Capitalisation
  - Numeric symbols
  - Punctuation marks
  - Position in the sentence
  - 14 features in total, combining above info, e.g., containsDigitAndDash (09-96), containsDigitAndComma (23,000.00)

IdentiFinder (2)

- Evaluation: MUC-6 (English) and MET-1(Spanish) corpora

- Mixed case English
  - IdentiFinder - 94.9% F-measure
  - Best rule-based – 96.4% F-measure
- Spanish mixed case
  - IdentiFinder – 90% F-measure
  - Best rule-based - 93% F-measure
  - Lower case names, noisy training data, less training data

- Impact of size of data- Trained with 650,000 words, but similar performance with half of the data. Less than 100,000 words reduce the performance to below 90% on English

MENE [Borthwick et al 98]

- Rule-based NE + ML based NE- achieve better performance
- Tokens tagged as: XXX_start, XXX_continue, XXX_end, XXX_unique, other (non-NE), where XXX is an NE category
- Uses Maximum Entropy (ME)
  - One only needs to find the best features for the problem
  - ME estimation routine finds the best relative weights for the features

MENE (2)

- Features
  - Binary features–"token begins with capitalised letter", "token is a four-digit number"
  - Lexical features–dependencies on the surrounding tokens (window ±2) e.g., "Mr" for people, "to" for locations
  - Dictionary features–equivalent to gazetteers (first names, company names, dates, abbreviations)
  - External systems–whether the current token is recognised as a NE by a rule-based system

MENE (3)

- MUC-7 formal run corpus
  - MENE – *84.2%* F-measure
  - Rule-based systems– *86% - 91 %* F-measure
  - MENE + rule-based systems – *92%* F-measure

- Learning curve
  - 20 docs – 80.97% F-measure
  - 40 docs – 84.14% F-measure
  - 100 docs – 89.17% F-measure
  - 425 docs – 92.94% F-measure

Named Entity Recognition: Maximum Entropy Approach Using Global Information

(*Chieu and Ng, 2003*)

Global Information

- Local Context is insufficient
  - "**Mary Kay** Names Vice Chairman…"
- Global Information is useful
  - "Richard C. Bartlett was named to the newly created position of vice chairman of **Mary Kay Corp**."

Named Entity Recognition

- Modeled as a classification problem

- Each token is assigned one of 29 (= 7*4 + 1) classes:
  - person_begin, person_continue, person_end, person_unique
  - org_begin, org_continue, org_end, org_unique,
  - …
  - nn (not-a-name)

Named Entity Recognition

Consuela Washington , a longtime

person_begin person_end nn nn nn

House staffer ... the Securities and

org_unique nn nn org_begin org_continue

Exchange Commission in the Clinton …

org_continue org_end nn nn person_unique

Maximum Entropy Modeling

The distribution p^* in the conditional ME framework:

$$p^*(s_i \mid s_{i-1}, o) = \frac{1}{Z(s_{i-1}, o)} \sum_a \exp(\alpha_a f_a(s_i, o))$$

$f_j(h,o)$: binary feature
α_j : parameter / weight of each feature

Java-based opennlp maxent package:
http://maxent.sourceforge.net

Checking for Valid Sequence

- To discard invalid sequences like:
  - person_begin location_end …
- Transition probability $P(c_i \mid c_{i-1}) = 1$ if a valid transition, 0 otherwise
  - Dynamic programming to determine the valid sequence of classes with highest probability

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

Local Features

- Case and zone
  - initCaps, allCaps, mixedCaps
  - TXT, HL, DATELINE, DD
- First word
- Word string
- Out-of-vocabulary
  - WordNet

Local Features

- InitCapPeriod (e.g., *Mr.*)
- OneCap (e.g., *A*)
- AllCapsPeriod (e.g., *CORP.*)
- ContainDigit (e.g., *AB3, 747*)
- TwoD (e.g., *99*)
- FourD (e.g., *1999*)
- DigitSlash (e.g., *01/01*)
- Dollar (e.g., *US$20*)
- Percent (e.g., *20%*)
- DigitPeriod (e.g., *$US3.20*)

Local Features

- Dictionary word lists
  - Person first names, person last names, organization names, location names
- Person prefix list (e.g., *Mr.*, *Dr.*), corporate suffix list (e.g., *Corp.*, *Inc.*)
  - Obtained from training data

- Month names, Days of the week, Numbers

Global Features

- Initcaps of other occurrences

Even Daily News have made the same mistake ….

They criticised **Daily News** for missing something **even** a boy would have noticed….

Global Features

- Person prefix and corporate suffix of other occurrences

Mary Kay Names Vice Chairman

Richard C. Bartlett was named to the newly created position of vice chairman of **Mary Kay Corp.**

Global Features

- Acronyms

The **Federal Communications Commission** killed

that plan last year … …

The company is still trying to challenge the **FCC**'s earlier decision … …

Global Features

- Sequence of initial caps

  [HL] First Fidelity Unit Heads Named

  [TXT] Both were executive vice presidents at First Fidelity.

NER – other approaches

- Hybrid systems
  - Combination of techniques
    - IBM's Intelligent Miner: Nominator + DB/2 data mining
  - WordNet hierarchies
    - MAGNINI et al. (2002)
  - Stacks of classifiers
    - Adaboost algorithm
  - Bootstrapping approaches
    - Small set of seeds
  - Memory-based ML, etc.

NER in various languages

- Arabic
  - TAGARAB (1998)
  - Pattern-matching engine + morphological analysis
  - Lots of morphological info (no differences in ortographic case)
- Bulgarian
  - OSENOVA & KOLKOVSKA (2002)
  - Handcrafted cascaded regular NE grammar
  - Pre-compiled lexicon and gazetteers
- Catalan
  - CARRERAS et al. (2003b) and MÁRQUEZ et al. (2003)
  - Extract Catalan NEs with Spanish resources (F-measure 93%)
  - Bootstrap using Catalan texts

NER in various languages

- Chinese & Japanese
  - Many works
  - Special characteristics
    - Character or word-based
    - No capitalization
  - CHINERS (2003)
    - Sports domain
    - Machine learning
    - Shallow parsing technique

NER in various languages

- ASAHARA & MATSMUTO (2003)
  - Character-based method
  - Support Vector Machine
  - 87.2% F-measure in the IREX (outperformed most word-based systems)
- Dutch
  - DE MEULDER et al. (2002)
    - Hybrid system
      - Gazetteers, grammars of names
      - Machine Learning Ripper algorithm

NER in various languages

- French
  - BÉCHET et al. (2000)
    - Decision trees
    - Le Monde news corpus
- German
  - Non-proper nouns also capitalized
  - THIELEN (1995)
    - Incremental statistical approach
    - 65% of corrected disambiguated proper names

NER in various languages

- Greek
  - KARKALETSIS et al. (1998)
    - English – Greek GIE (Greek Information Extraction) project
    - GATE platform
- Italian
  - CUCCHIARELLI et al. (1998)
    - Merge rule-based and statistical approaches
    - Gazetteers
    - Context-dependent heuristics
    - ECRAN (Extraction of Content: Research at Near Market)
    - GATE architecture
    - Lack of linguistic resources: 20% of NEs undetected

NER in various languages

- Korean
  - CHUNG et al. (2003)
    - Rule-based model, Hidden Markov Model, boosting approach over unannotated data

- Portuguese
  - SOLORIO & LÓPEZ (2004, 2005)
    - Adapted CARRERAS et al. (2002b) spanish NER
    - Brazilian newspapers

NER in various languages

- Serbo-croatian
  - NENADIC & SPASIC (2000)
    - Hand-written grammar rules
    - Highly inflective language
      - Lots of lexical and lemmatization pre-processing
    - Dual alphabet (Cyrillic and Latin)
      - Pre-processing stores the text in an independent format
- Spanish
  - CARRERAS et al. (2002b)
    - Machine Learning, AdaBoost algorithm
    - BIO and Open Close approaches

NER in various languages

- Swedish
  - SweNam system (DALIANIS & ASTROM, 2001)
    - Perl
    - Machine Learning techniques and matching rules

- Turkish
  - TUR et al (2000)
    - Hidden Markov Model and Viterbi search
    - Lexical, morphological and context clues

Named Entity Recognition

- Multilingual approaches
  - Goals - CUCERZAN & YAROWSKY (1999)
    - To handle basic language-specific evidences
    - To learn from small NE lists (about 100 names)
    - To process large and small texts
    - To have a good class-scalability (to allow the definition of different classes of entities, according to the language or to the purpose)
    - To learn incrementally, storing learned information for future use

Named Entity Recognition

- Multilingual approaches
  - GALLIPI (1996)
    - Machine Learning
    - English, Spanish, Portuguese
  - ECRAN (Extraction of Content: Research at Near Market)
  - REFLEX project (2005)
    - the US National Business Center

Named Entity Recognition

- Multilingual approaches
  - POIBEAU (2003)
    - Arabic, Chinese, English, French, German, Japanese, Finnish, Malagasy, Persian, Polish, Russian, Spanish and Swedish
    - UNICODE
    - Language independent architecture
    - Rule-based, machine-learning
    - Sharing of resources (dictionary, grammar rules…) for some languages
  - BOAS II (2004)
    - University of Maryland Baltimore County
    - Web-based
    - Pattern-matching
    - No large corpora

NER – other topics

- Character vs. word-based
  - JING et al. (2003)
    - Hidden Markov Model classifier
    - Character-based model better than word-based model
- NER translation
  - Cross-language Information Retrieval (CLIR), Machine Translation (MT) and Question Answering (QA)
- NER in speech
  - No punctuation, no capitalization
  - KIM & WOODLAND (2000)
    - Up to 88.58% F-measure
- NER in Web pages
  - wrappers

NER in Indian Languages

Problems for NER 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
- 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

NER in Indian Languages

- LI and McCallum (2004)-Hindi
  - CRF model using feature induction technique to automatically construct the features
  - Features:
    - Word text, character n-grams (n=2, 3, 4), word prefix and suffix of lengths 2,3,4
    - 24 Hindi gazetteer lists
    - Features at the current, previous and next sequence positions were made available
  - Dataset: 601 BBC and 27 EMI Hindi documents
  - Performance
    - *F-measure* of 71.5% with an early stopping point of 240 iterations of L-BFGS for the 10-fold cross validation experiments

NER in Indian Languages

- Saha et al. (2008)-Hindi
  - ME model
  - Features:
    - Statistical and linguistic feature sets
    - Hindi gazetteer lists
    - Semi-automatic induction of context patterns
    - Context patterns as features of the MaxEnt method
  - Dataset: 243K words of Dainik Jagaran (training)
    25K (test)
  - Performance
    - *F-measure* of 81.52%

NER in Indian Languages

- Patel et al. (2008)-Hindi and Marathi
  - Inductive Logic Programming (ILP) based techniques for automatically extracting rules for NER from tagged corpora and background knowledge
  - Dataset: 54340 (Marathi), 547138 (Hindi)
  - Performance
    - *PER:* 67%, *LOC:* 71% *and ORG:* 53% (Hindi)
    - *PER:* 82%, *LOC:* 48% *and ORG:* 55% (Hindi)
  - Advantages over rule-based system
    - development time reduces by a factor of 120 *compared to a linguist doing the entire rule development*
    - *a complete and consistent view of all significant patterns in the data at the level of abstraction*

NER in Indian Languages

- Ekbal and Saha (2011)-Bengali, Hindi, Telugu and Oriya
  - Genetic algorithm based weighted ensemble
  - Classifiers: ME, CRF and SVM
  - Features:
    - Word text, word prefix and suffix of lengths 1,2,3; PoS
    - Context information, various orthographic features etc.
  - Dataset: Bengali (Training: 312,947; Test: 37,053)

    Hindi (Training: 444,231; Test: 58,682)

    Telugu (Training: 57,179; Test: 4,470)

    Oriya (Training: 93,573; Test: 2,183)
  - Performance
    - *F-measures: Bengali* (92.15%), *Hindi* (92.20%), *Telugu* (84.59%) and *Oriya* (89.26%)

NER in Indian Languages

- Ekbal and Saha (2012)-Bengali, Hindi and Telugu
  - Multiobjective Genetic algorithm based weighted ensemble
  - Classifiers: ME, CRF and SVM
  - Features:
    - Word text, word prefix and suffix of lengths 1,2,3; PoS
    - Context information, various orthographic features etc.
  - Dataset: Bengali (Training: 312,947; Test: 37,053)

    Hindi (Training: 444,231; Test: 58,682)

    Telugu (Training: 57,179; Test: 4,470)

    Oriya (Training: 93,573; Test: 2,183)
  - Performance
    - *F-measures: Bengali* (92.46%), *Hindi* (93.20%), *Telugu* (86.54%)

NER in Indian Languages

- Shishtla et al. (2008)- Telugu and Hindi
  - CRF
  - Character-n gram approach is more effective than word-based model
  - Features
    - Word-internal features, PoS, chunk etc.
    - No external resources

  -Datasets: Telugu (45,714 tokens); Hindi ((45,380 tokens)

  -Performance

    - F-measures: Telugu (49.62%), Hindi (45.07%)

NER in Indian Languages

- Vijayakrishna and Sobha (2008)
  - CRF
  - Tourism domain with 106 hierarchical tags
  - Features
    - Roots of words, PoS, dictionary of NEs, patterns of certain types of NEs (date, time, money etc.) etc
  - Performance
    - 80.44%

NER in Indian Languages

- Saha et al. (2008)- Hindi
  - Maximum Entropy
  - Features
    - Statistical and linguistics features
    - Word clustering
    - Clustering used for feature reduction in Maximum Entropy
- -Datasets: 243K Hindi newspaper "Dainik Jagaran".

  -Performance
    - F-measures: 79.03% (approximately 7% improvement with Clusters)

Other works in Indian Languages NER

- Gali et al. (2008)-Bengali, Hindi, Telugu and Oriya
  - CRF
- Kumar and Kiran (2008)-Bengali, Hindi, Telugu and Oriya
  - CRF
- Srikanth and Murthy (2008) –Telugu
  - CRF
- Goyal (2008)-Hindi
  - CRF
- Nayan et al. (2008)-Hindi
  - Phonetic matching technique

Other works in Indian Languages NER

- Ekbal et al. (2008)-Bengali
  - CRF
- Saha et al. (2009)-Hindi
  - Semi-supervised approach
- Saha et al. (2010)-Hindi
  - SVM with string based kernel function
- Ekbal and Saha (2010)-Bengali, Hindi and Telugu
  - GA based classifier ensemble selection
- Ekbal and Saha (2011)-Bengali, Hindi and Telugu
  - Multiobjective simulated annealing approach for classifier ensemble

Other works in Indian Languages NER

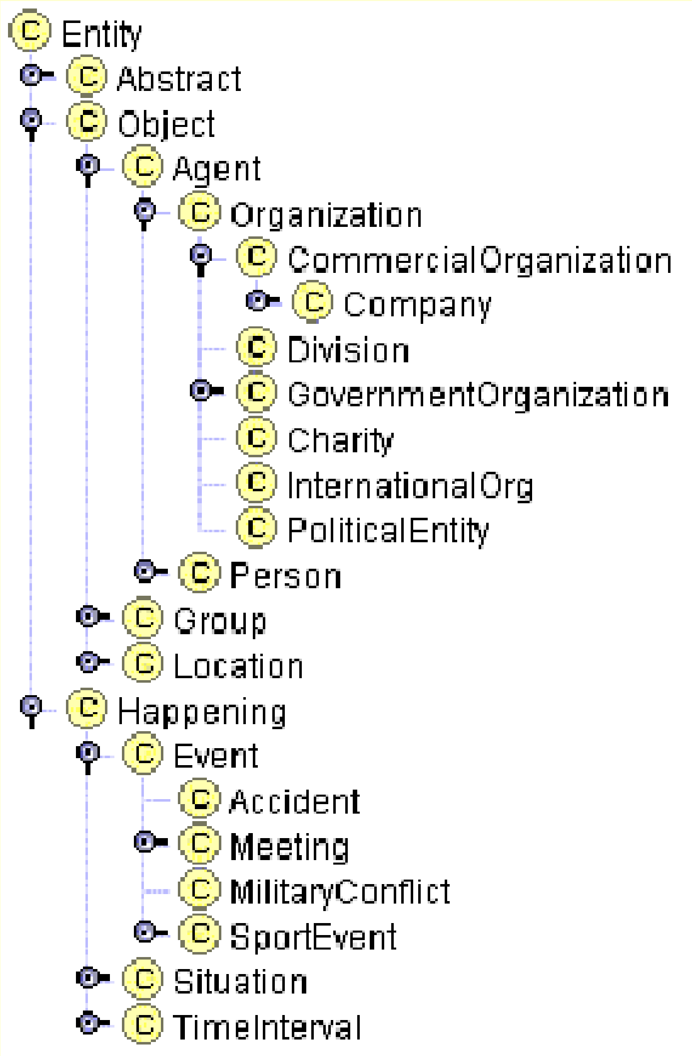
- Saha et al. (2012)-Hindi and Bengali
  - Comparative techniques for feature reductions
- Ekbal and Saha (2012)-Bengali, Hindi and Telugu
  - Multiobjective approach for feature selection and classifier ensemble
- Ekbal et al. (2012)-Hindi and Bengali
  - Active learning
  - Effective in a resource-constrained environment

Shared Tasks on Indian Language NER

- NERSSEAL Shared Task- 2008 (http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=2)
- NLPAI ML Contest 2007- (http://ltrc.iiit.ac.in/nlpai_contest07/cgi-bin/index.cgi)

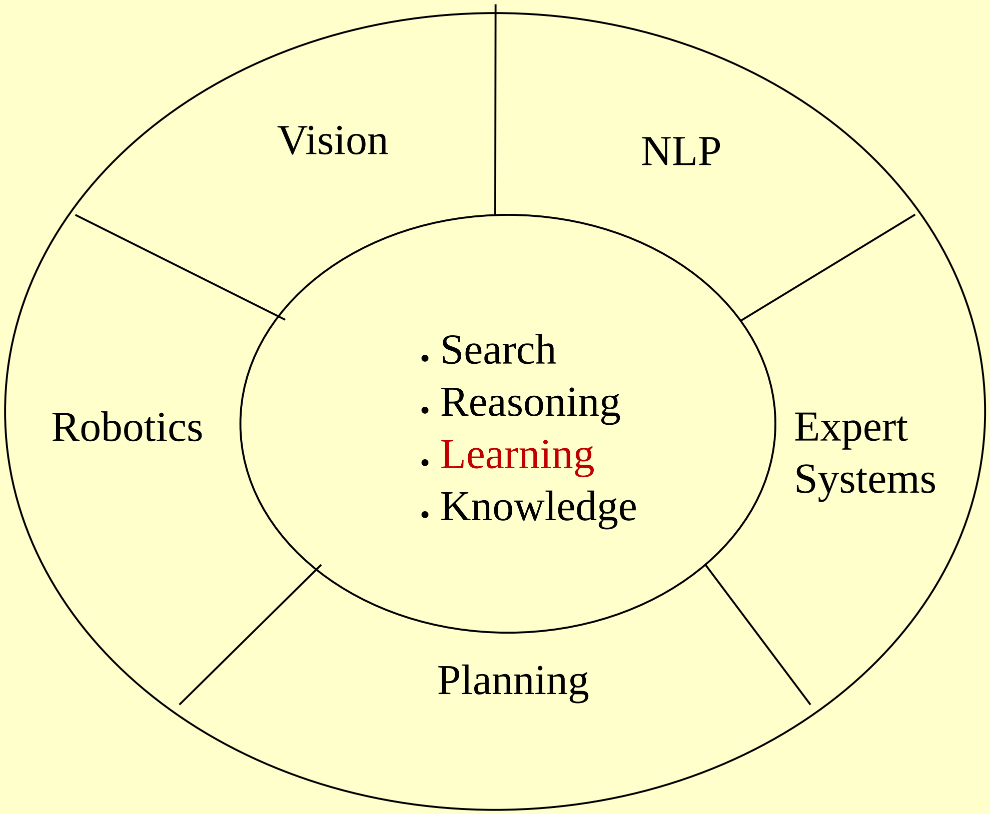
Evaluating Richer NE Tagging

- Hierarchy/ontology-based NE tagging
- Need to take into account distance in the hierarchy
- Tagging a company as a charity is less wrong than tagging it as a person



Machine Learning: A very brief introduction

AI: The various Components



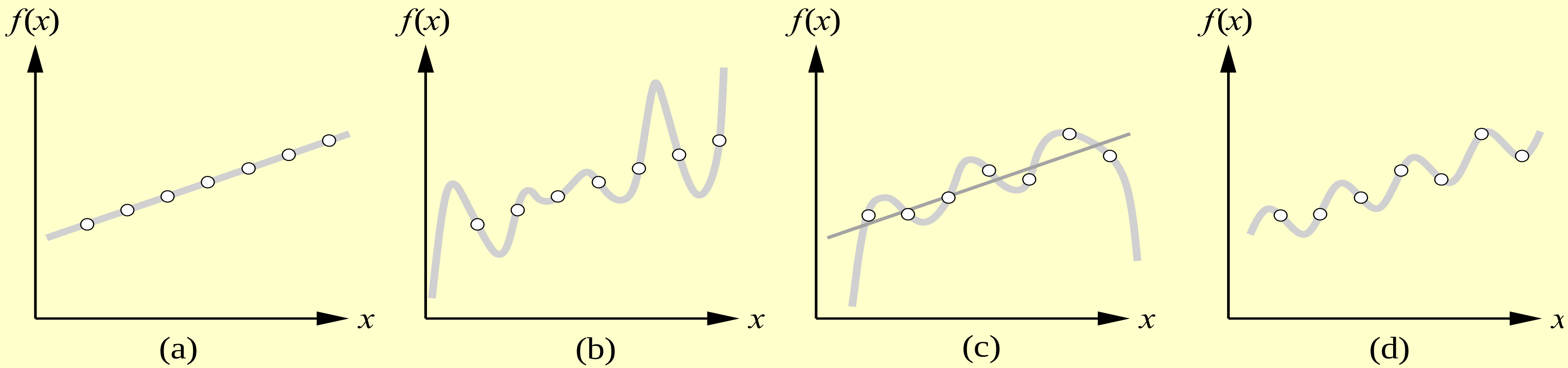
Machine Learning

- Machine learning: how to acquire a model on the basis of data / experience
  - Learning parameters (e.g. probabilities)
  - Learning structure (e.g. BN graphs)
  - Learning hidden concepts (e.g. clustering)

Machine Learning

- Unsupervised Learning
  - No feedback from teacher; detect patterns
- Reinforcement Learning
  - Feedback consists of rewards/punishment
- Supervised Learning
  - Examples of correct answers are given
  - Discrete answers: *Classification*
  - Continuous answers: *Regression*

Supervised Machine Learning



Given a training set:

$(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots (x_n, y_n)$

Where each y_i was generated by an unknown $y = f(x)$,
Discover a function h that approximates the true function f

Example: Spam Filter

- Input: x = email
- Output: y = "spam" or "ham"
- Setup:
  - Get a large collection of example emails, each labeled "spam" or "ham"
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future emails
- Features: The attributes used to make the ham / spam decision
  - Words: FREE!
  - Text Patterns: $dd, CAPS
  - Non-text: SenderInContacts
  - …

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

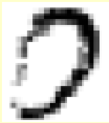
99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

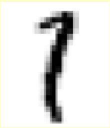
Example: Digit Recognition

- Input: x = images (pixel grids)
- Output: y = a digit 0-9
- Setup:
  - Get a large collection of example images, each labeled with a digit
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future digit images

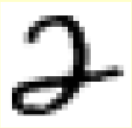
- Features: The attributes used to make the digit decision
  - Pixels: (6,8)=ON
  - Shape Patterns: NumComponents, AspectRatio, NumLoops
  - …



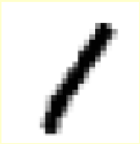
0



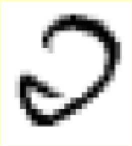
1



2



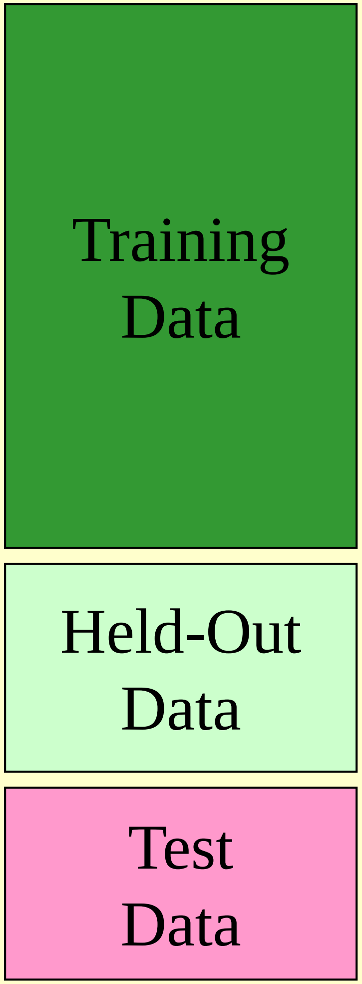
1



??

How to Learn

- Data: labeled instances, e.g. emails marked spam/ham
  - Training set
  - Held out (validation) set
  - Test set
- Features: attribute-value pairs which characterize each x
- Experimentation cycle
  - Learn parameters (e.g. model probabilities) on training set
  - Tune hyperparameters on held-out set
  - Compute accuracy on test set
  - Very important: never "peek" at the test set!
- Evaluation
  - Accuracy: fraction of instances predicted correctly
- Overfitting and generalization
  - Want a classifier which does well on *test* data
  - Overfitting: fitting the training data very closely, but not generalizing well to test data



Categorization/Classification

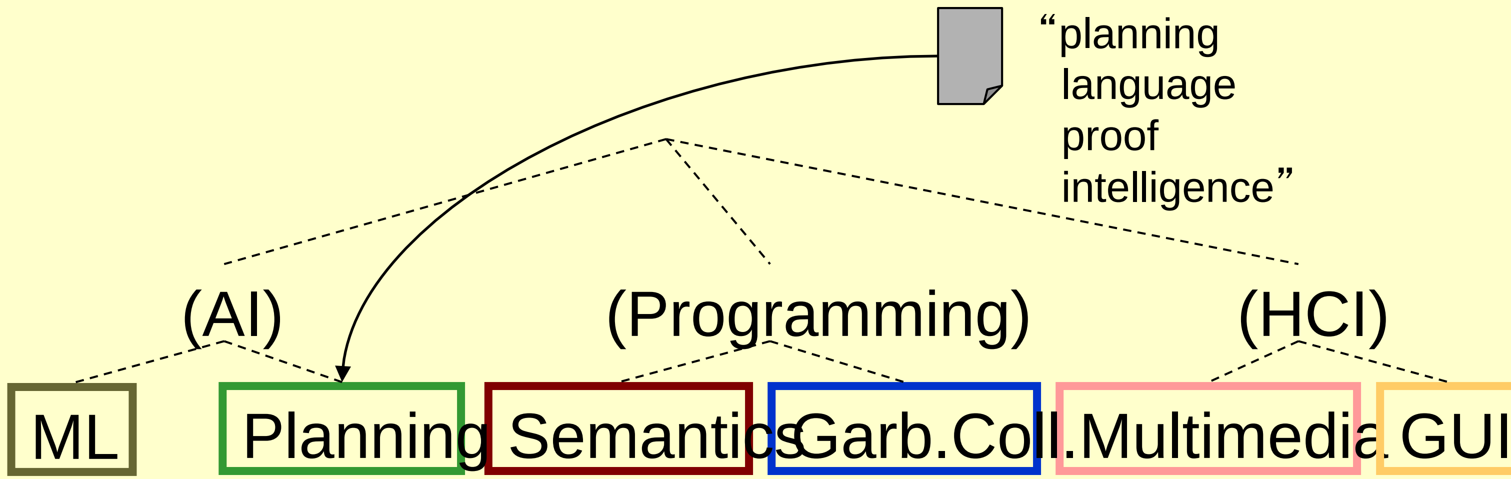
- Given:
  - A description of an instance, $d \in X$
    - X is the *instance language* or *instance space*
      - Issue: how to represent text documents?
      - Usually some type of high-dimensional space
  - A fixed set of classes:

    $C = \{c_1, c_2, \ldots, c_J\}$
- Determine:
  - The category of d: $\gamma(d) \in C$, where $\gamma(d)$ is a *classification function* whose domain is X and whose range is C
    - We want to know how to build classification functions ("classifiers")

Document Classification

Test Data:

Classes:



Training Data:

ML	Planning	Semantics	Garb.Coll.	Multimedia	GUI
learning	planning	programming	garbage	...	...
intelligence	temporal	semantics	collection		
algorithm	reasoning	language	memory		
Reinforcement	plan	proof...	optimization		
network...	language...		region...		

Classification Methods (1)

- Manual classification
  - Used by the original Yahoo! Directory
  - Looksmart, about.com, ODP, PubMed
  - Very accurate when job is done by experts
  - Consistent when the problem size and team both are small
  - Difficult and expensive to scale
    - Means we need automatic classification methods for big problems

Classification Methods (2)

- Automatic classification
  - Hand-coded rule-based systems
    - One technique used by Reuters, CIA, etc.
    - It's what Google Alerts is doing
      - Widely deployed in government and enterprise
    - Companies provide "IDE" (integrated development environment) for writing such rules
    - E.g., assign category if document contains a given boolean combination of words
    - Standing queries: Commercial systems have complex query languages (everything in IR query languages +score accumulators)
    - Accuracy is often very high if a rule has been carefully refined over time by a subject expert
    - Building and maintaining these rules is expensive
    - Rules could vary with the change of domain

Classification Methods (3)

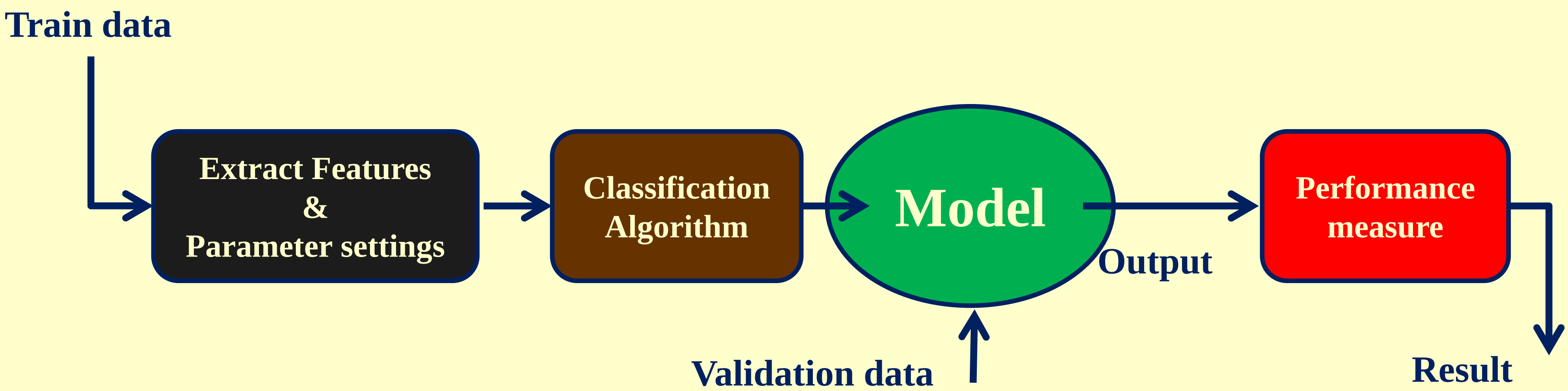
- *Supervised learning of a document*-label assignment function
  - Many systems partly rely on machine learning (Autonomy, Microsoft, Enkata, Yahoo!, Google News, …)
    - k-Nearest neighbors (simple, powerful)
    - Naive Bayes (simple, common method)
    - Support-vector machines (new, more powerful)
    - … plus many other methods
    - Requirement: requires hand-classified training data
    - But data can be built up (and refined) by amateurs

- Many commercial systems use a mixture of methods

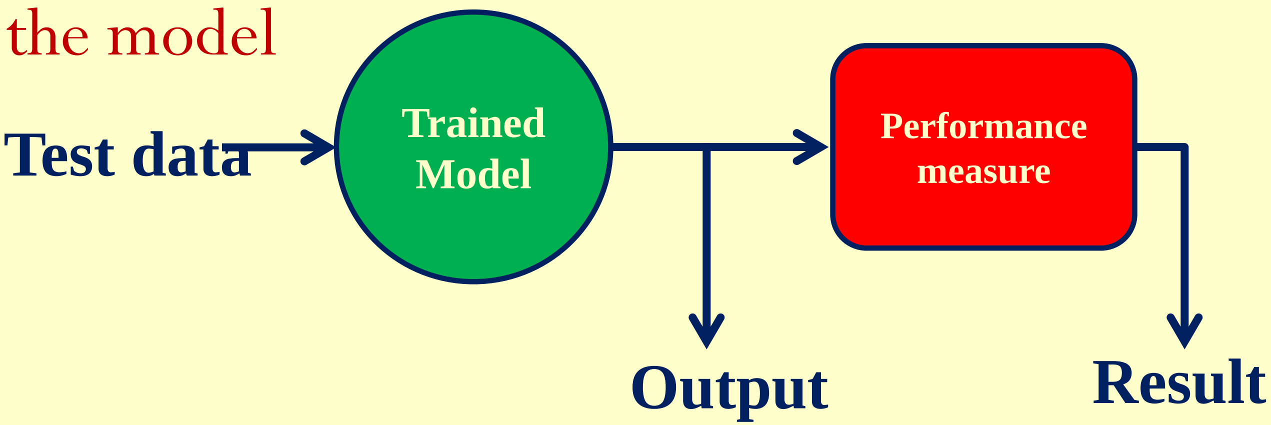
- And the recent trend is deep learning
    - Automatically learns feature on its own
    - Has received significant attention to the researchers of computer vision, and very recently to NLP

Machine Learning

- Training a model



- Testing the model



HMM based NERC

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 $P (T \mid W)$

$$T' = \mathrm{argmax}_T \, P (T \mid W)$$

HMM based NERC System (Contd..)

By Bayes' Rule,

$$P(T \mid W) = P(W \mid T) * P(T) / P(W)$$

$$T' = \mathrm{argmax}_T \, P(W \mid T) * P(T)$$

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

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

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

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

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

Emission Probability: $P(w_i \mid t_i) = \frac{freq(w_i, t_i)}{freq(t_i)}$

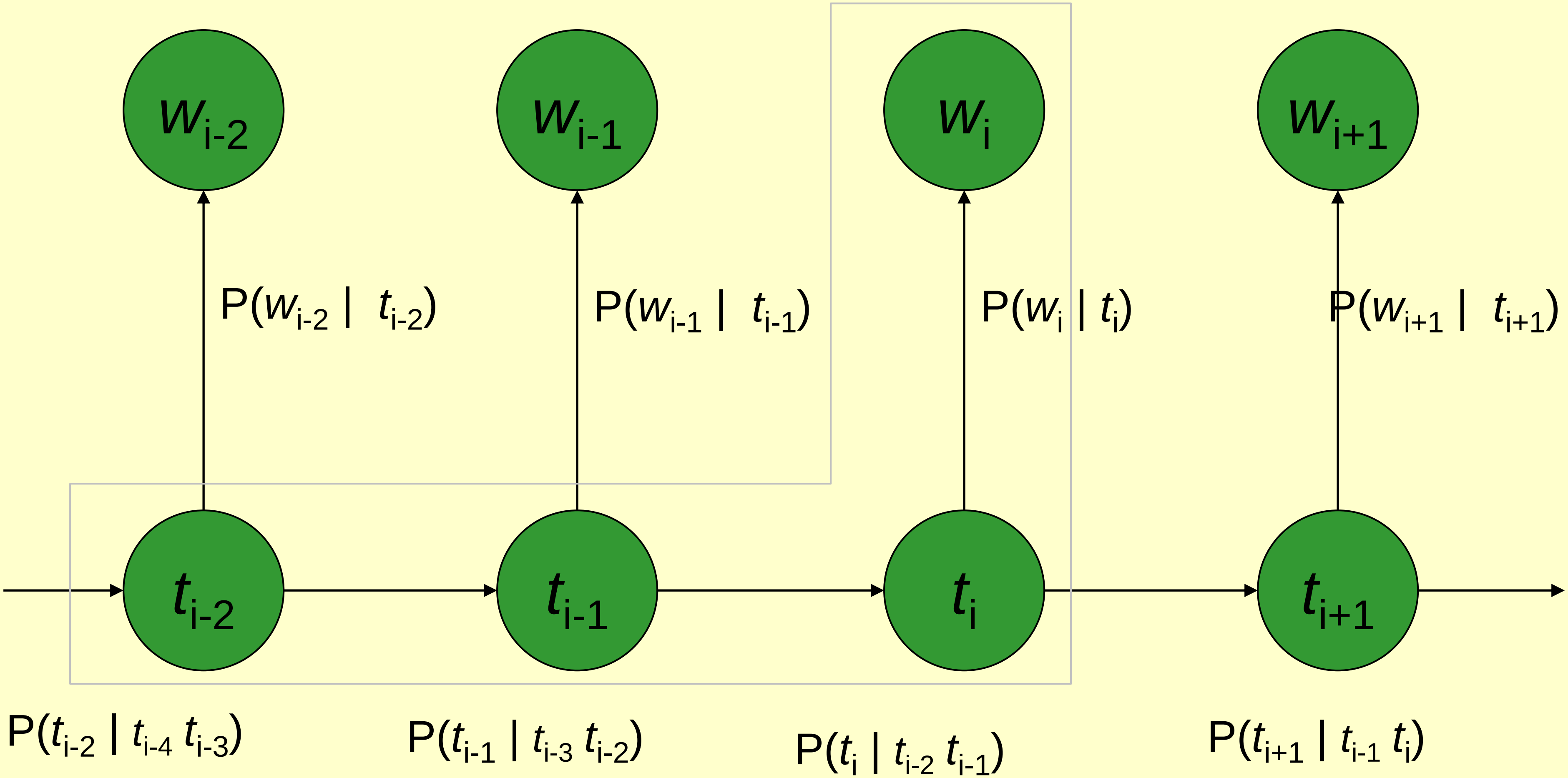
HMM based NERC System (Contd..)

- Context Dependency (Our Modification)
  - Markov model is made more powerful by introducing 1st order context dependent feature

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

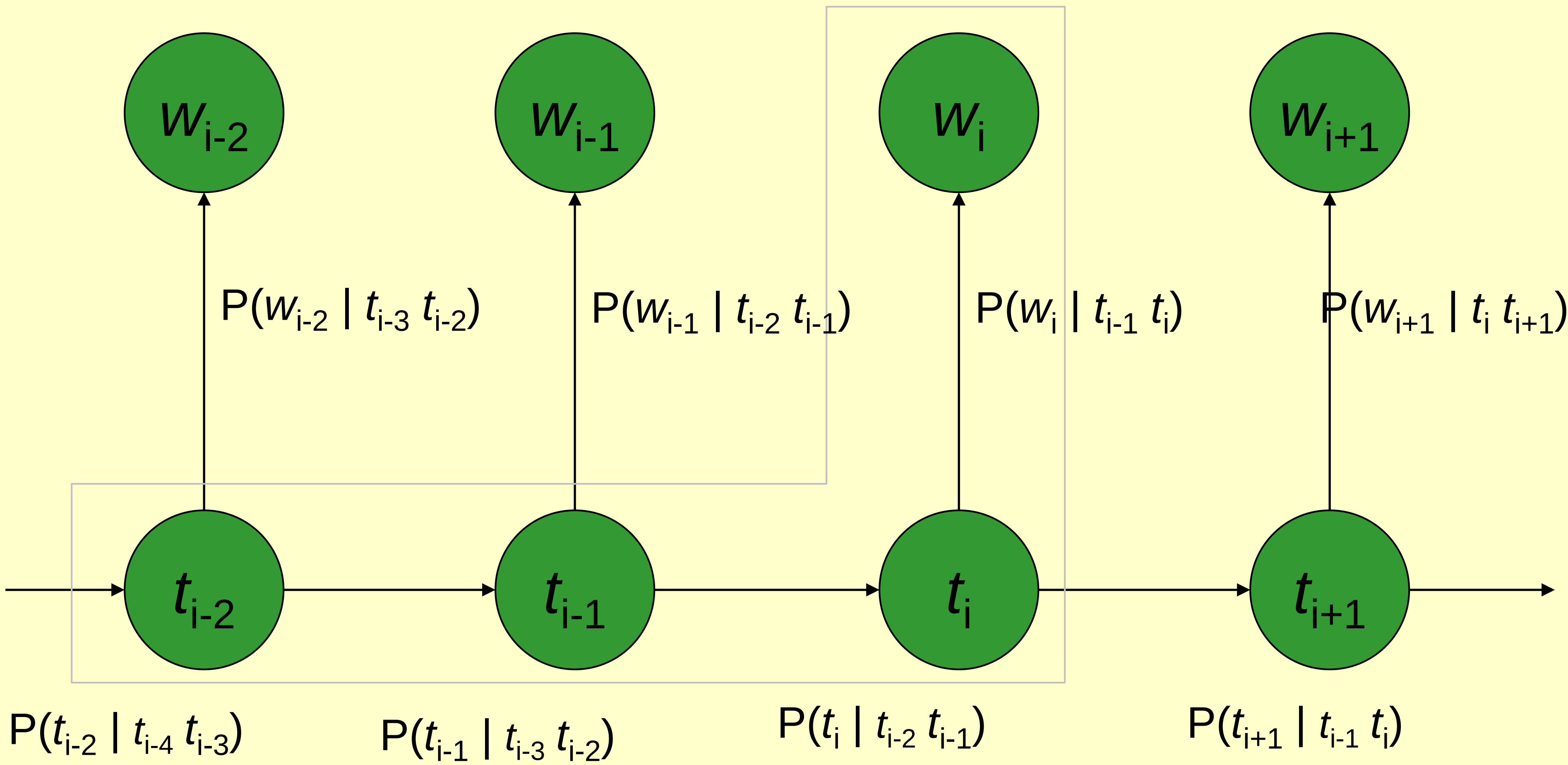
$$P(w_i \mid t_{i-1}, t_i) = \frac{freq(t_{i-1}, t_i, w_i)}{freq(t_{i-1}, t_i)}$$

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?
  - 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 (*Linear Interpolation*)
  - Transition probability

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

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

  - Emission probability

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

$$\theta_1 + \theta_2 = 1$$

  - Calculation of λs and Θs (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)

Results of the HMM based System: Bengali

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

Ensemble Learning: A brief Introduction

Drawbacks of Single Classifier

- The "best" classifier not necessarily the ideal choice

- For solving a classification problem, many individual classifiers with different parameters are trained
  - The "best" classifier will be selected according to some criteria e.g., *training accuracy* or *complexity of the classifiers*

- Problems: Which one is the best?
  - Maybe more than one classifiers meet the criteria (e.g. same training accuracy), especially in the following situations:

    -Without sufficient training data

    – Learning algorithm leads to different local optima easily

Drawbacks of Single Classifier

- Potentially valuable information may be lost by discarding the results of less-successful classifiers

  E.g., the discarded classifiers may correctly classify some samples

- Other drawbacks
  - Final decision must be wrong if the output of selected classifier is wrong
  - Trained classifier may not be complex enough to handle the problem

Ensemble Learning

- Employ multiple learners and combine their predictions
- Methods of combination:
  - Bagging, boosting, voting
  - Error-correcting output codes
  - Stacked generalization
  - Cascading
  - …
- **Advantage:** improvement in predictive accuracy
- **Disadvantage:** it is difficult to understand an ensemble of classifiers

Why Do Ensembles Work?

Dietterich(2002) showed that ensembles overcome three problems:

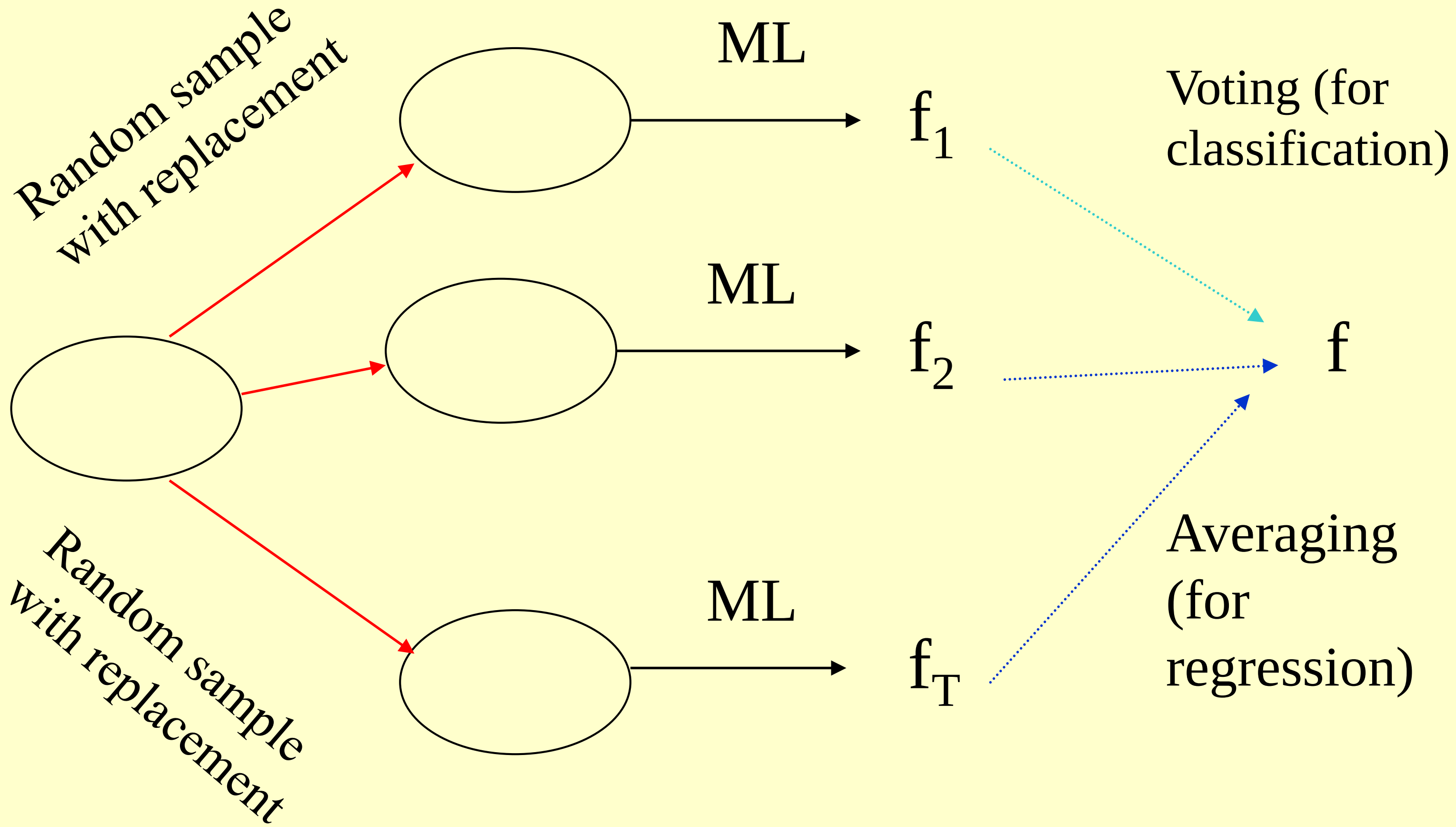
- ***Statistical Problem-*** arises when the hypothesis space is too large for the amount of available data. Hence, there are many hypotheses with the same accuracy on the data and the learning algorithm chooses only one of them! There is a risk that the accuracy of the chosen hypothesis is low on unseen data!
- ***Computational Problem-*** arises when the learning algorithm cannot guarantee finding the best hypothesis.
- ***Representational Problem-*** arises when the hypothesis space does not contain any good approximation of the target class(es).

T.G. Dietterich, Ensemble Learning, 2002

Categories of Ensemble Learning

- Methods for Independently Constructing Ensembles
  - Bagging
  - Randomness Injection
  - Feature-Selection Ensembles
  - Error-Correcting Output Coding
- Methods for Coordinated Construction of Ensembles
  - Boosting
  - Stacking
  - Co-training

Bagging (**B**ootstrap **Agg**regration)



Bagging is only effective when using **unstable** (i.e. a small change in the training set can cause a significant change in the model) nonlinear models

Randomization Injection

- Inject some randomization into a standard learning algorithm (usually easy):
  - Neural network: random initial weights
  - Decision tree: when splitting, choose one of the top N attributes at random (uniformly)

- Dietterich (2000) showed that 200 randomized trees are <u>statistically significantly</u> better than C4.5 for over 33 datasets!

Feature-Selection Ensembles (Random Subspace Method)

- ***Key idea:*** Provide a different subset of the input features in each call of the learning algorithm

- ***Example:*** Venus&Cherkauer (1996) trained an ensemble with 32 neural networks. The 32 networks were based on 8 different subsets of 119 available features and 4 different algorithms. The ensemble was significantly better than any of the neural networks!

Error-correcting output codes

- Very elegant method of transforming multi-class problem into two-class problem
  - Simple scheme: as many binary class attributes as original classes using one-per-class coding

class	class vector
a	1000
b	0100
c	0010
d	0001

- Train f(ci) for each bit
- Idea: use *error-correcting codes* instead

Error-correcting output codes

- Example:

class	class vector
a	1111111
b	0000111
c	0011001
d	0101010

  - What's the true class if base classifiers predict 1011111?

ECOC-more.ppt

Dietterich, Ghulum Bakiri.
Journal of Artificial Intelligence Research 2 1995. Solving **Multiclass Learning Problems via**. **Error-Correcting Output Codes.**

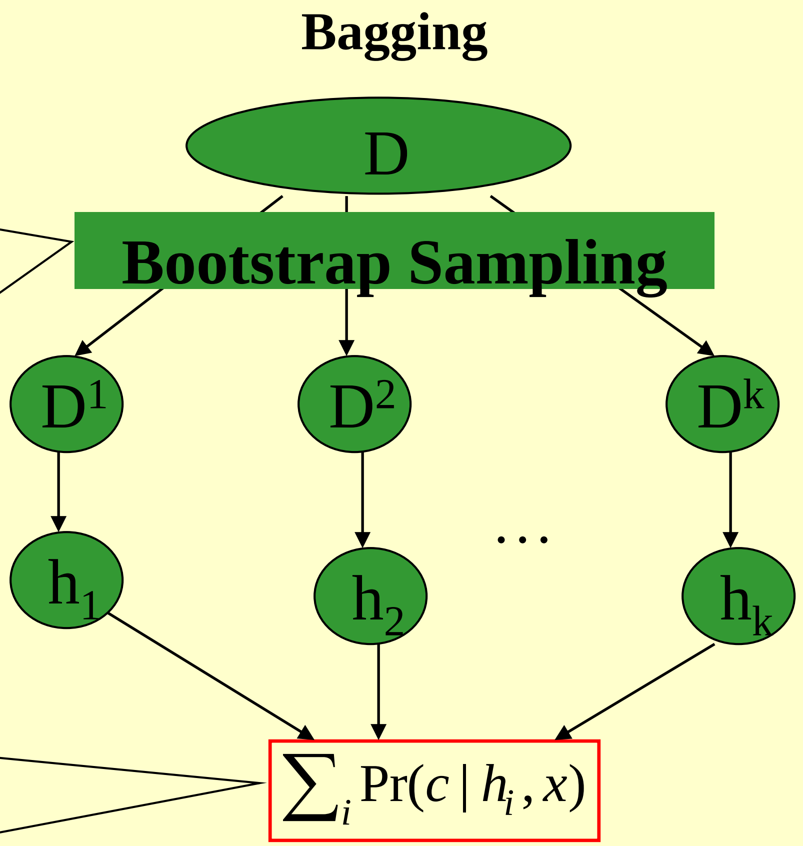
Methods for Coordinated Construction of Ensembles

- ***Key idea-*** to learn *complementary* classifiers so that instance classification is realized by taking a weighted sum of the classifiers:
  - Boosting
  - Stacking

Inefficiency with Bagging

Inefficiency with bootstrap sampling:

- Every example has equal chance to be sampled
- No distinction between “easy” examples and “difficult” examples



Inefficiency with model combination

- A constant weight for each classifier
- No distinction between accurate classifiers and inaccurate classifiers

Improve the Efficiency of Bagging

- Better sampling strategy
  - Focus on the examples that are difficult to classify correctly

- Better combination strategy
  - Accurate model should be assigned with more weights

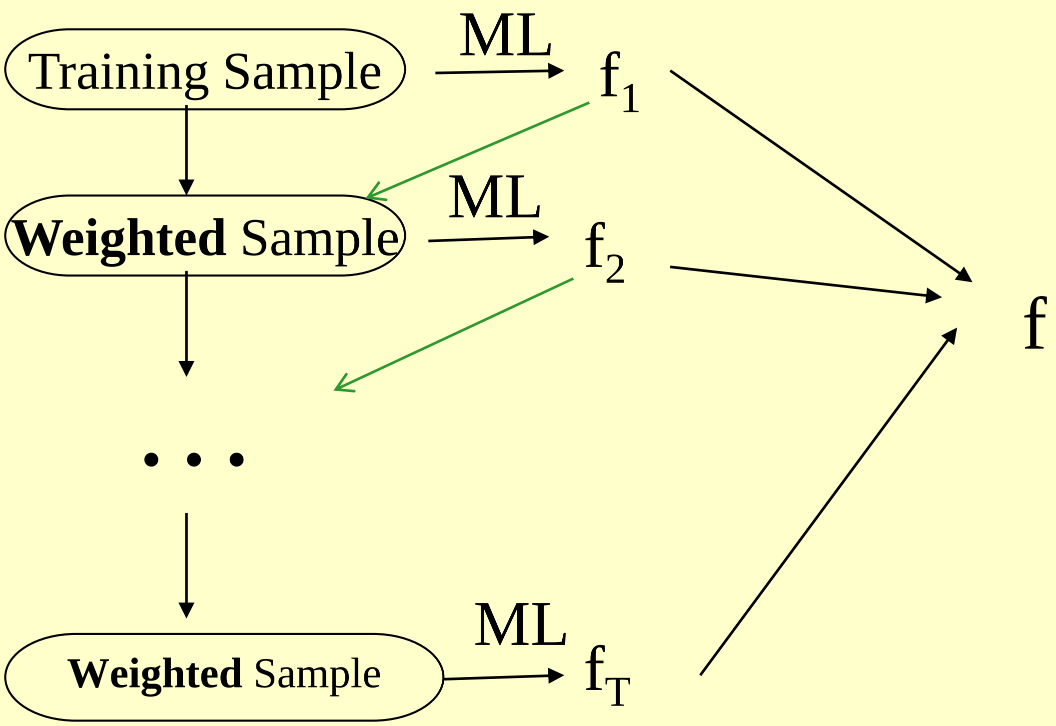
Overview of Boosting

- Introduced by Schapire and Freund in 1990s
- "Boosting": convert a weak learning algorithm into a strong one
- Main idea: Combine many weak classifiers to produce a powerful committee
- Algorithms:
  - **AdaBoost**: adaptive boosting
  - Gentle AdaBoost
  - BrownBoost
  - …

Boosting

- Uses voting/averaging but models are weighted according to their performance
- Iterative procedure: new models are influenced by performance of previously built ones
  - New model encouraged to become expert for instances classified incorrectly by earlier models
  - Intuitive justification: *models should be experts that complement each other*
- Several variants of this algorithm exist!

Boosting



Boosting: Use the same sample with different weights to generate classifiers

Bagging: Use different samples with identical weights to generate classifiers

Strengths of AdaBoost

- No parameters to tune (except for the number of rounds)
- Fast, simple and easy to program (??)
- Comes with a set of theoretical guarantee (e.g., *training error*, *test error*)

- Instead of trying to design a learning algorithm that is accurate over the entire space, we can focus on finding base learning algorithms that only need to be better than random

- Can identify outliers: i.e. examples that are either mislabeled or inherently ambiguous and hard to categorize

Weakness of AdaBoost

- Actual performance depends on the data and the base learner

- Boosting seems to be especially susceptible to **noise**

- When the number of outliers is very large, the emphasis placed on the hard examples can hurt the performance

  ➔ "Gentle AdaBoost", "BrownBoost"

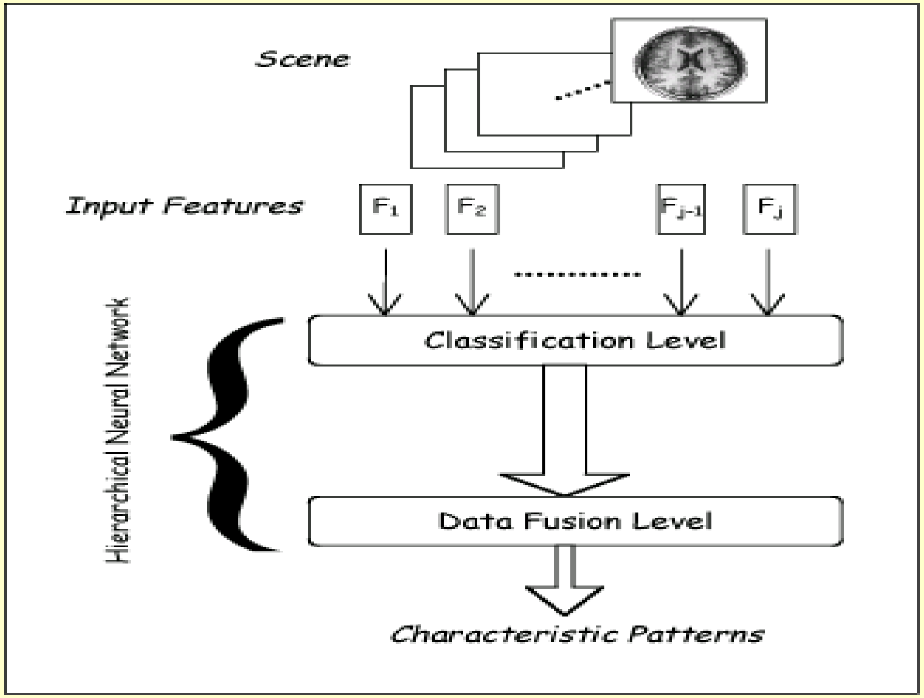
Comparison of Bagging and Boosting

- Bagging always uses *re-sampling* rather than *re-weighting*
- Bagging does not modify the distribution over examples or mislabels, but instead always uses the uniform distribution
- In forming the final hypothesis, bagging gives equal weight to each of the weak hypotheses

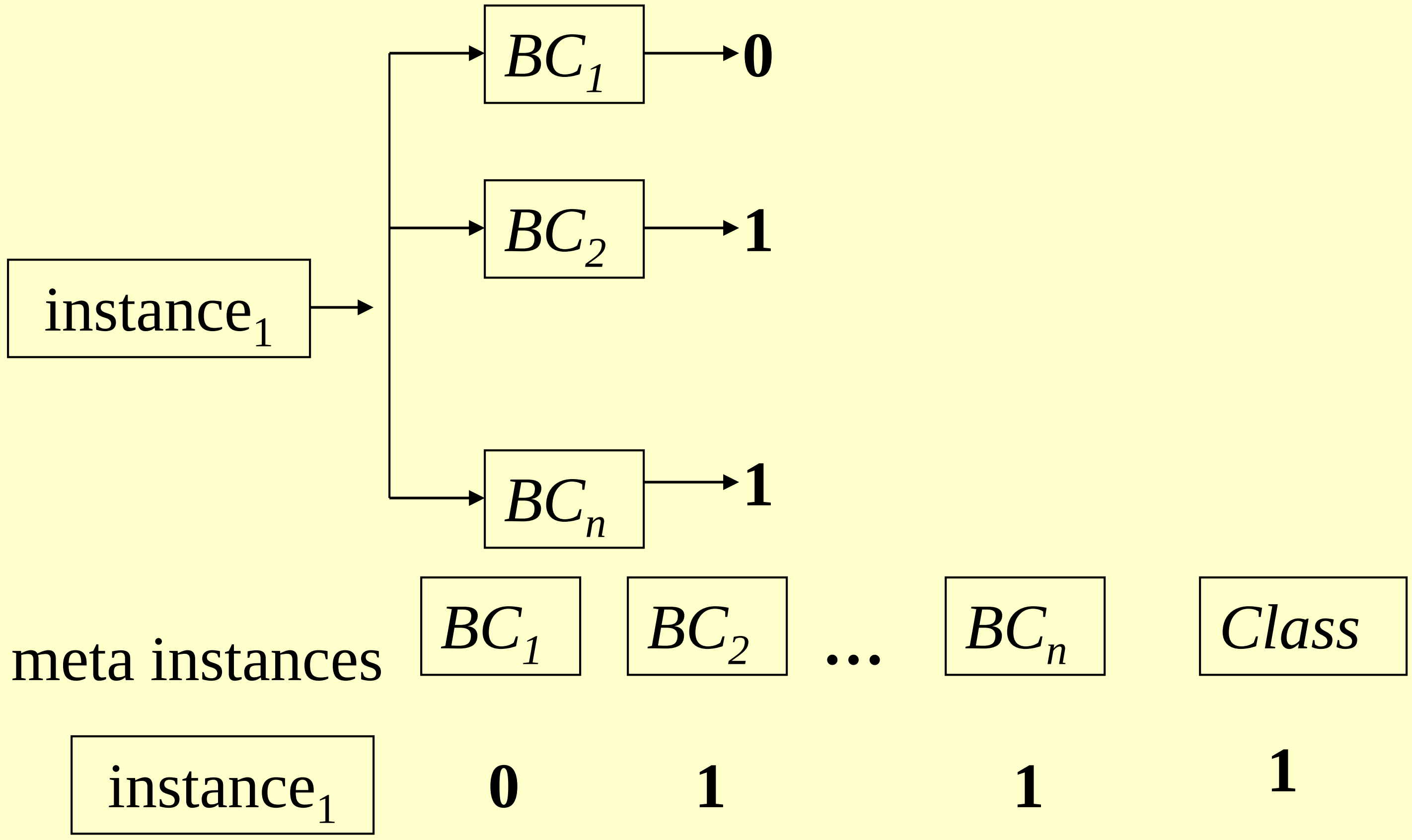
Stacking

- Uses *meta learner* instead of voting to combine predictions of base learners
  - Predictions of base learners (*level-0 models*) are used as input for meta learner (*level-1 model)*
- Base learners- usually different learning schemes

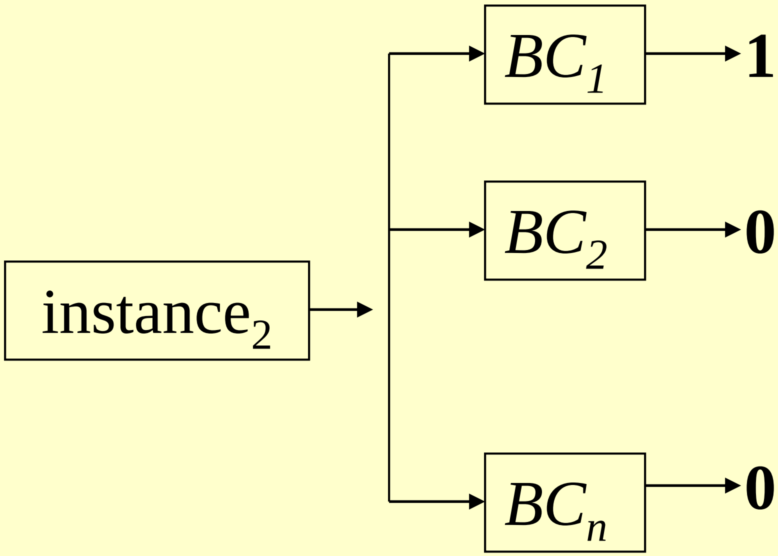
Hierarchical Neural Networks



Stacking


instance$_1$
BC_1
0
BC_2
1
BC_n
1
meta instances
BC_1
BC_2
...
BC_n
Class
instance$_1$
0
1
1
1

Stacking



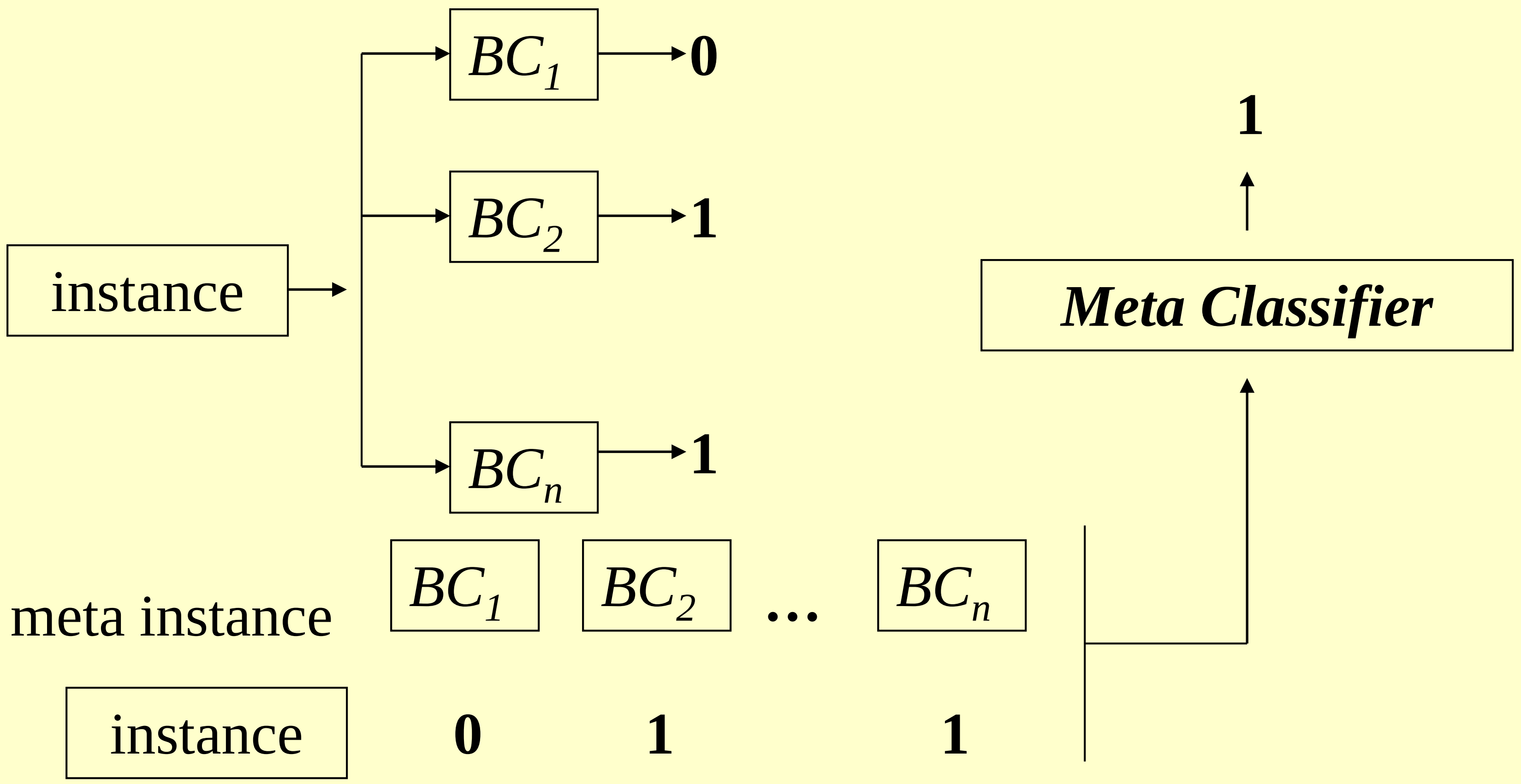
meta instances	BC_1	BC_2	…	BC_n	$Class$
instance$_1$	**0**	**1**		**1**	**1**
instance$_2$	**1**	**0**		**0**	**0**

Stacking

Meta Classifier

meta instances	BC_1	BC_2	…	BC_n	*Class*
instance_1	**0**	**1**		**1**	**1**
instance_2	**1**	**0**		**0**	**0**

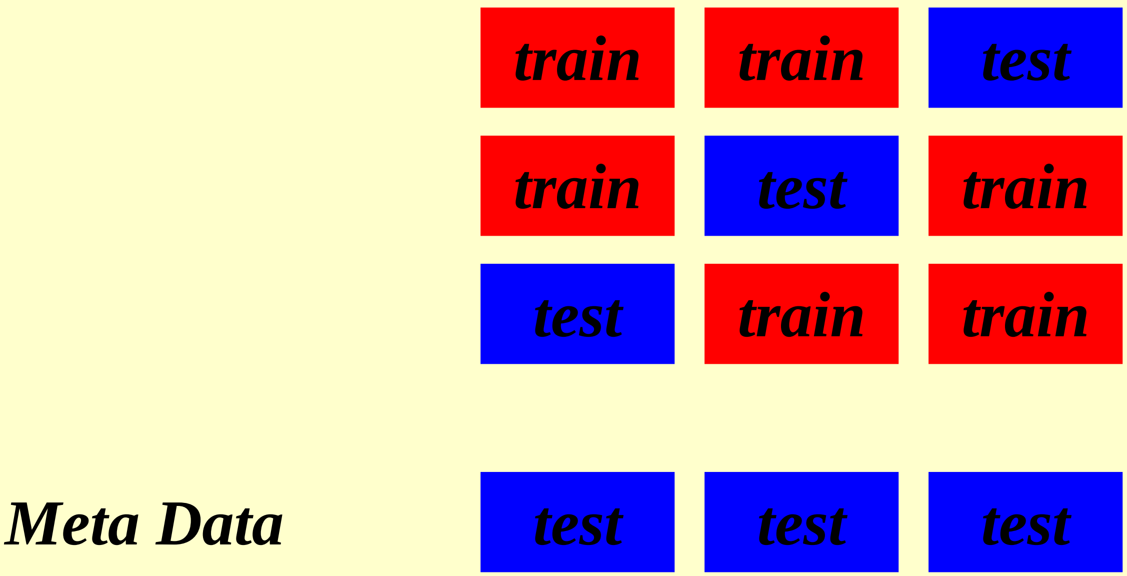
Stacking



instance
BC1
BC2
BCn
0
1
1
1
Meta Classifier
meta instance
BC1
BC2
…
BCn
instance
0
1
1

More on stacking

- Predictions on training data can't be used to generate data for level-1 model! The reason is that the level-0 classifier that better fits training data will be chosen by the level-1 model! Thus,
- k-fold cross-validation-like scheme is employed! An example for k = 3!



Some Practical Advices

- If the classifier is **unstable** (i.e, decision trees) then apply bagging!
- If the classifier is **stable and simple** (e.g. Naïve Bayes) then apply boosting!
- If the classifier is **stable and complex** (e.g. Neural Network) then apply randomization injection!
- If you have many classes and a binary classifier then try error-correcting codes! If it does not work then use a complex binary classifier!

Evolutionary Algorithms for Classifier Ensemble

Evolutionary Algorithms in NLP

- Good Review (L. Araujo, 2007)
- Natural language tagging- Alba, G. Luque, and L. Araujo (2006)
- Grammar Induction-T. C. Smith and I. H. Witten (1995)
- Phrase-structure-rule of natural language-W. Wang and Y. Zhang (2007)
- Information retrieval-R. M. Losee (2000)
- Morphology -D. Kazakov (1997)
- Dialogue systems-D. Kazakov (1998)
- Grammar inference -M. M. Lankhors (1994)
- Memory-based language processing (A. Kool, W. Daelemans, and J. Zavrel., 2000)

Evolutionary Algorithms in NLP

- Anaphora resolution:Veronique Hoste (2005), Ekbal et al. (2011), Saha et al. (2012)
- Part-of-Speech tagging: Araujo L (2002)
- Parsing: Araujo L (2004)
- Document clustering: Casillas A et al. (2003)
- Summarization: Andersson L (2004)
- Machine Translation : Jun Suzuki (2012)
- NER: Ekbal and Saha (2010; 2011; 2012 etc.)

Genetic Algorithm: Quick Overview

- Randomized search and optimization technique
- Evolution produces good individuals, similar principles might work for solving complex problems
- Developed: USA in the 1970' s by J. Holland
- Got popular in the late 1980's
- Early names: J. Holland, K. DeJong, D. Goldberg
- Based on ideas from *Darwinian Evolution*
- Can be used to solve a variety of problems that are not easy to solve using other techniques

Genetic Algorithm: Similarity with Nature

Genetic Algorithms	←→	Nature
A solution (phenotype)		Individual
Representation of a solution (*genotype*)		Chromosome
Components of the solution		Genes
Set of solutions		Population
Survival of the fittest (*Selection*)		Darwins theory
Search operators		Crossover and mutation
Iterative procedure		Generations

Basic Steps of Genetic Algorithm

1. $t = 0$
2. initialize population $P(t)$ /* $Popsize = |P|$ */
3. for $i = 1$ to $Popsize$
   compute fitness $P(t)$
4. $t = t + 1$
5. if termination criterion achieved go to step 10
6. select (P)
7. crossover (P)
8. mutate (P)
9. go to step 3
10. output best chromosome and stop

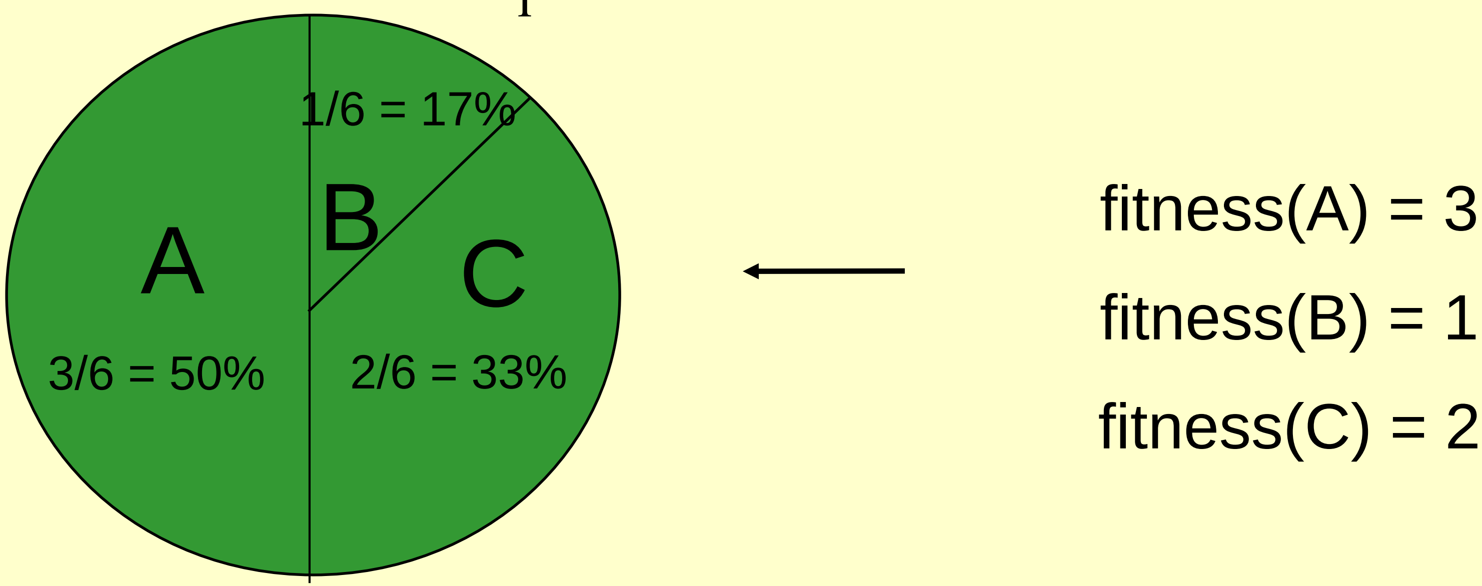
End

Example population

No.	Chromosome	Fitness
1	1010011010	1
2	1111100001	2
3	1011001100	3
4	1010000000	1
5	0000010000	3
6	1001011111	5
7	0101010101	1
8	1011100111	2

GA operators: Selection

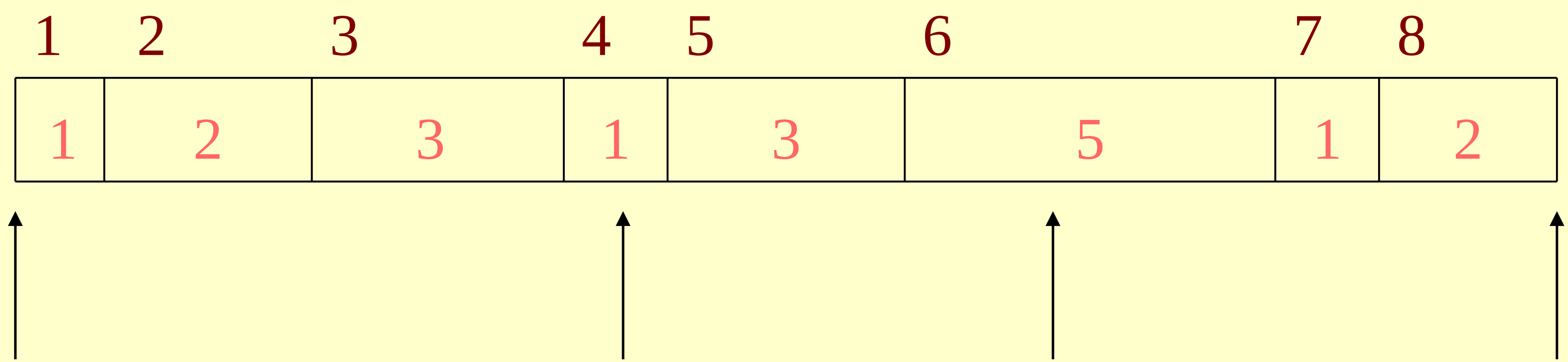
- Main idea: better individuals get higher chance
  - Chances proportional to fitness
  - Implementation: roulette wheel technique
    - » Assign to each individual a part of the roulette wheel
    - » Spin the wheel n times to select n individuals



GA operator: Selection

- Add up the fitness's of all chromosomes
- Generate a random number R in that range
- Select the first chromosome in the population that - when all previous fitness's are added including the current one- gives you at least the value R

Roulette Wheel Selection



0

Rnd[0..18] = 7 Rnd[0..18] = 12

18

Chromosome 4 Chromosome 6

Parent1 Parent2

GA operator: Crossover

- Choose a random point on the two parents
- Split parents at this crossover point
- With some high probability (*crossover rate*) apply crossover to the parents
  - P_c typically in range (0.6, 0.9)
- Create children by exchanging tails

Crossover - Recombination

1010000000	Parent1	Offspring1	1011011111
1001011111	Parent2	Offspring2	1000000000

Crossover single point - random

Single Point Crossover

n-point crossover

- Choose n random crossover points
- Split along those points
- Glue parts, alternating between parents
- Generalisation of 1 point (still some positional bias)

Mutation

mutate

Original offspring		Mutated offspring	
Offspring1	1011011111	Offspring1	1011001111
Offspring2	1010000000	Offspring2	1000000000

Original offspring

Mutated offspring

With some small probability (the *mutation rate*) flip each bit in the offspring (*typical values between 0.1 and 0.001*)

A. Ekbal and S. Saha (2011). Weighted Vote-Based Classifier Ensemble for Named Entity Recognition: A Genetic Algorithm-Based Approach. ACM Transactions on Asian Language Information Processing (ACM TALIP), Vol. 2(9),

DOI=10.1145/1967293.1967296

http://doi.acm.org/10.1145/1967293.1967296

Weighted Vote based Classifier Ensemble

- Motivation
  - All classifiers are not equally good at detecting all the classes

- Weighted voting: weights of voting vary among the classes for each classifier
  - *High*: Classes for which the classifier perform good
  - *Low*: Classes for which it's output is not very reliable
- *Crucial issue*: Selection of appropriate weights of votes per classifier

Problem Formulation

Let *no. of classifiers*=N, and *no. of classes*=M

Find the weights of votes V per classifier optimizing a function F(V)

-V: an real array of size N × *M*

-V(i , j) : weight of vote of the *i*th classifier for the *j*th class

-V(i , j) ε [0, 1] denotes the degree of confidence of the *i*th classifier for the *j*th class

maximize F(B) ;

F ε {*recall, precision, F-measure*} and B is a subset of A

Here, *F1= F-measure*

Chromosome representation

0.59 | 0.12 | 0.56 | 0.09 |0.91 | 0.02 | 0.76 | 0.5 | 0.21

Classifier-1 Classifier-2 Classifier-3

- Real encoding used
- Entries of chromosome randomly initialized to a real (r) between 0 and 1: r = rand () / RAND_MAX+1
- If the population size P then all the P number of chromosomes of this population are initialized in the above way

Fitness Computation

Step-1: For M classifiers, F_i i= 1 to M be the F-measure values

Step-2: Train each classifier with 2/3 training data and evaluate with the remaining 1/3 part

Step-3: For ensemble output of the 1/3 test data, apply weighted voting on the outputs of M classifiers

(a). Weight of the output label provided by the *mth* classifier = I (m, i)

Here, *I(m, i) is the entry of the chromosome corresponding to mth* classifier and *ith class*

(b). Combined score of a class for a word *w*

Fitness Computation

Op(w, m): output class produced by the *mth* classifier for word *w*

Class receiving the maximum score selected as joint decision

Step-4: Compute overall F-measure value for 1/3 data

Step-5: Steps 3 and 4 repeated to perform 3-fold cross validation

Step-6: Objective function or fitness function = F-$measure_{avg}$

Objective: Maximize the objective function using search capability of GA

Other Parameters

- Selection
  - Roulette wheel selection (*Holland, 1975; Goldberg, 1989*)
- Crossover
  - Normal Single-point crossover (Holland, 1975)
- Mutation
  - Probability selected adaptively (*Srinivas and Patnaik, 1994*)
  - Helps GA to come out from local optimum

Termination Condition

- Execute the processes of *fitness computation*, *selection*, *crossover*, and *mutation* for a maximum number of generations
- *Best solution*-Best string seen up to the last generation

- Best solution indicates
  - Optimal voting weights for all classes in each classifier

- Elitism implemented at each generation
  - Preserve the best string seen up to that generation in a location outside the population
  - Contains the most suitable classifier ensemble

NE Features: Mostly language independent

NE 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)

NE Features

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

NE Features

- 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

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

NE Features

- 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.

NE Features

- Part of Speech (POS) Information: POS tag(s) of the current and/or the surrounding word(s)
  - SVM-based POS tagger (Ekbal and Bandyopadhyay, 2008)
  - SVM based NERC→POS tagger developed with a fine-grained tagset of 27 tags
  - Coarse-grained POS tagger
    - Nominal, PREP (Postpositions) and Other

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

Datasets

- Web-based Bengali news Corpus (Ekbal and Bandyopadhyay, 2008, *Language Resources and Evaluation of Springer*)
  - *34 million* wordforms
  - News data collection of 5 years
- NE annotated corpus for Bengali
  - Manually annotated 250K wordforms
  - IJCNLP-08 Shared Task on NER for South and South East Asian Languages (available at http://ltrc.iiit.ac.in/ner-ssea-08)
- NE annotated datasets for Hindi and Telugu
  - NERSSEAL shared task

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

Training and Test Datasets

Language	#Words in training	#NEs in training	#Words in test	#NEs in test
Bengali	312,947	37,009	37,053	4,413
Hindi	444,231	26,432	32,796	58,682
Telugu	57,179	4,470	6,847	662
Oriya	93,573	4,477	2,183	206

Experiments

- Classifiers used
  - Maximum Entropy (ME): Java based OpenNLP package (http://maxent.sourceforge.net/)
  - Conditional Random Field: C++ based CRF++ package (http://crfpp.sourceforge.net/)
  - Support Vector Machine:
    - YamCha toolkit (http://chasen-org/ taku/software/yamcha/)
    - TinySVM-0.07 (http://cl.aist-nara.ac.jp/ taku-ku/software/TinySVM)
    - Polynomial kernel function

Experiments

- GA: population size=50, number of generations=40, mutation and crossover probabilities are selected adaptively.

- Baselines
  - Baseline 1: Majority voting of all classifiers
  - Baseline 2: Weighted voting of all classifiers (*weight*: overall average F-measure value)
  - Baseline 3: Weighted voting of all classifiers (*weight*: F-measure value of the individual class)

Results (*Bengali*)

Model	Recall	Precision	F-measure
Best Individual Classifier	89.42	90.55	89.98
Baseline-1	84.83	85.90	85.36
Baseline-2	85.25	86.97	86.97
Baseline-3	86.97	87.34	87.15
Stacking	90.17	91.74	90.95
ECOC	89.78	90.89	90.33
QBC	90.01	91.09	90.55
GA based ensemble	92.08	92.22	92.15

Results (*Hindi*)

Model	Recall	Precision	F-measure
Best Individual Classifier	88.72	90.10	89.40
Baseline-1	63.32	90.99	74.69
Baseline-2	74.67	94.73	83.64
Baseline-3	75.52	96.13	84.59
Stacking	89.80	90.61	90.20
ECOC	90.16	91.11	90.63
GA based ensemble	96.07	88.63	92.20

Results (*Telugu*)

Model	Recall	Precision	F-measure
Best Individual Classifier	77.42	77.99	77.70
Baseline-1	60.12	87.39	71.23
Baseline-2	71.87	92.33	80.33
Baseline-3	72.22	93.10	81.34
Stacking	77.65	84.12	80.76
ECOC	77.96	85.12	81.38
GA based ensemble	78.82	91.26	84.59

Results (*Oriya*)

Model	Recall	Precision	F-measure
Best Individual Classifier	86.55	88.03	87.29
Baseline-1	86.95	88.33	87.63
Baseline-2	87.12	88.50	87.80
Baseline-3	87.62	89.12	88.36
Stacking	87.90	89.53	88.71
ECOC	87.04	88.56	87.79
GA based ensemble	88.56	89.98	89.26

Results (*English*)

Model	Recall	Precision	F-measure
Best Individual Classifier	86.16	85.24	86.31
Baseline-1	85.75	86.12	85.93
Baseline-2	86.20	87.02	86.61
Baseline-3	86.65	87.25	86.95
Stacking	85.93	86.45	86.18
ECOC	86.12	85.34	85.72
GA based ensemble	88.72	88.64	88.68

Current Trends in NE Research

- Development of domain-independent and language-independent systems
  - Can be easily portable to different domains and languages
- Fine-grained NE classification
  - May be at the hierarchy of WordNet
  - Beneficial to the fine-grained IE
  - Helps in Ontology learning

Current Trends in NE Research

- NER systems in non-newswire domains
  - Humanities (arts, history, archeology, literature etc.): *lots of non-traditional entities are present*
  - Chemical and bio-chemical (*long and nested NEs*)
  - Biomedical texts and clinical records (*long and nested NEs; does not follow any standard nomenclature)*
  - Unstructured datasets such as Twitter, online product reviews, blogs, SMS etc.

Study Materials: References

- ***Named Entities: Recognition, Classification and Use, Special Issue of Lingvisticae Investigationes Journal***, Satoshi Sekine and Elisabete Ranchhod (Eds.), Vol. 30:1 (2007), John Benjamins Publishing Company
- All relevant conferences- ACL, COLING, EACL, IJCNLP, CiCLing , AAAI, ECAI etc.
- Named Entities Workshop (NEWS)
- Biotext Mining challenges- BioCreative, BioNLP etc.
- NER in unstructured text: NER in twitter (*ACL 2015 and COLING 2016 Shared Tasks*), NER in code-mixed data (*Fire shared task-16*)

Important Resources

- Stanford NER: Classifier: CRF; Language: English; Types: PER, LOC and ORG
- LingPipe: Hybrid; News Entities: PER, LOC and ORG; Biomedical: Genes, Organisms, Chemicals
- TextPro: Supervised SVM (YamCha); Languages: Italian, English and German; Entities: PER, LOC and ORG
- GATE: Hybrid System; Language: English; Entities: PER, LOC and ORG
- BANNER: Classifier: CRF; Entities: Gene and Gene Products
- GENIA Tagger: HMM; Entities: Protein, DNA, RNA, Cell_Line and Cell_Type
- Important Datasets: CoNLL 2002/2003, JNLPBA-2004, BioCreative, IJCNLP-08 NERSSEAL, Twitter NER (W-NUT 2016/15)

NERC in Biomedical Domain

Aims: Text mining

- *Data Mining* -> needs structured data, usually in numerical form
- *Text mining*: discover & extract unstructured knowledge hidden in text–Hearst (1999)
- Text mining aids to construct hypotheses from associations derived from text
  - protein-protein interactions
  - associations of genes–*phenotypes*
  - functional relationships among genes…etc.

An Example

- *Stress is associated with migraines*
- *Stress can lead to loss of magnesium*

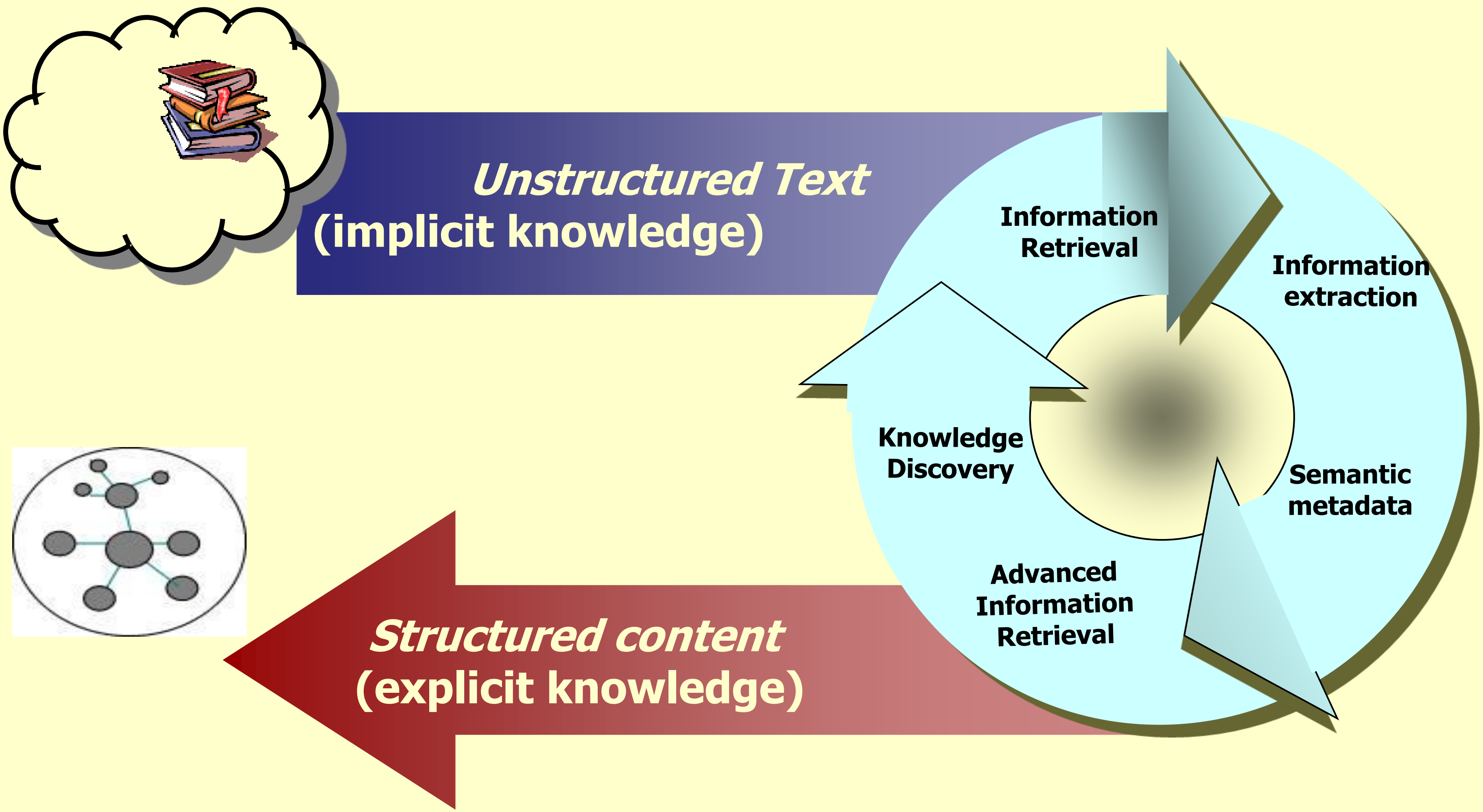
=> *Loss of magnesium may cause migraine*

Text Mining in biomedicine

- Why biomedicine?
  - Consider just MEDLINE: 23,000,000 references, 40,000-50,000 added per month
  - Dynamic nature of the domain: new terms (*genes*, *proteins*, *chemical compounds*, *drugs* etc.) constantly created
  - Impossible to manage such an information overload

From Text to Knowledge:

tackling the data deluge through text mining



Reading

- Book on BioTextMining
  - S. Ananiadou & J. McNaught (eds) (2006). Text Mining for Biology and Biomedicine, ArtechHouse
  - McNaught, J. & Black, W. (2006) Information Extraction, Text Mining for Biology & Biomedicine, Artechhouse, pp.143-177
- Detailed bibliography in Bio-Text Mining
  - BLIMPhttp://blimp.cs.queensu.ca/
  - http://www.ccs.neu.edu/home/futrelle/bionlp/

Bio-textmining Campaigns

Some biotext mining campaigns

- KDD Cup-2002
- TREC-Genomics (http://ir.ohsu.edu/genomics/)
- JNLPBA-2004 (http://www.nactem.ac.uk/tsujii/GENIA/ERtask/report.html): Named entity recognition
- BioCreative (www.biocreative.org)-Information extraction including NER, PPI, text categorization etc. (2004, 2006, 2008,2010,2011, 2012, 2013, 2014, 2015, 2016, 2017 etc.)
- BioNLP 2009, 2011, 2013, 2015-detailed biological phenomenon

  (http://www.nactem.ac.uk/tsujii/GENIA/SharedTask

Method: *Weighted vote based classifier Ensemble (already discussed)*

NE Extraction in Biomedicine

- Objective-identify biomedical entities and classify them into some predefined categories
  - *E.g. Protein*, *DNA*, *RNA*, *Cell_Line*, *Cell_Type*
- *Major Challenges*
  - building a complete dictionary for all types of biomedical NEs is ***infeasible due to the generative nature of NEs***
  - NEs are made of *very long compounded words* (i.e., contain nested entities) or abbreviations and hence difficult to classify them properly
  - names do not follow any nomenclature

Challenges (Contd..)

- NEs include different symbols, common words and punctuation symbols, conjunctions, prepositions etc.
  - NE boundary identification is more difficult and challenging
- Same word or phrase can refer to different NEs based on their contexts

Features: Domain-Independent

- Context Word: Preceding and succeeding words
- Word Suffix and Prefix
    - Fixed length character strings stripped from the ending or beginning of word
- Class label: Class label(s) of the previous word (s)
- Length (binary valued): Check whether the length of the current word less than three or not (shorter words rarely NEs)
- Infrequent (binary valued): Infrequent words in the training corpus most probably NEs

Features

- Part of Speech (PoS) information- PoS of the current and/or surrounding token(s)
  - GENIA tagger V2.0.2 (http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger)
- Chunk information-Chunk of the current and/or surrounding token(s)
  - GENIA tagger V2.0.2
- Unknown token feature-checks whether current token appears in training

Features

- Word normalization
  - feature attempts to reduce a word to its stem or root form (from GENIA tagger O/P)

- Head nouns
  - major noun or noun phrase of a NE that describes its function or the property
  - E.g. *factor* is the head noun for the NE *NF-kappa B transcription factor*

Features

- Verb trigger
  - Special types of verbs (e.g., *binds*, *participates* etc.)
  - Occurs preceding to NEs
  - Provides useful information about the NE class

- Word class feature-Certain kinds of NEs, which belong to the same class, are similar to each other
  - Capital letters→ A, small letters→a, number→O and non-English characters→-
  - Consecutive same characters are squeezed into one character
  - Groups similar names into the same NE class

Features

- Informative words
  - NEs are too *long, complex* and contain *many common words* that are actually not NEs
  - Function words- *of*, *and* etc.; nominals such as *active*, *normal* etc. appear in the training data often more frequently but these don't help to recognize NEs
  - Informative words extracted from the training data

- Content words in surrounding contexts-*Exploits global context information*

Features

- *Orthographic Features*-defined based on the construction of words

Feature	Example	Feature	Example
InitCap	Src	AllCaps	EBNA, LMP
InCap	mAb	CapMixAlpha	NFkappaB, EpoR
DigitOnly	1, 123	DigitSpecial	12-3
DigitAlpha	2× NFkappaB, 2A	AlphaDigitAlpha	IL23R, EIA
Hyphen	-	CapLowAlpha	Src, Ras, Epo
CapsAndDigits	32Dc13	RomanNumeral	I, II
StopWord	at, in	ATGCSeq	CCGCCC, ATAGAT
AlphaDigit	p50, p65	DigitCommaDigit	1,28
GreekLetter	alpha, beta	LowMixAlpha	mRNA, mAb

Experiments

- **Datasets**-JNLPBA 2004 shared task datasets
  - **Training**: 2000 MEDLINE abstracts with 500K wordforms
  - **Test**: 404 abstracts with 200K wordforms
- **Tagset**: 5 classes
  - Protein, DNA, RNA, Cell_line, Cell_type
- **Classifiers**
  - CRF and SVM

- **Evaluation scheme**: JNLPBA 2004 shared task script (http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/ERtask/report.html)
  - Recall, precision and F-measure according to *exact boundary match*, *right* and *left* boundary matching

Experiments

Model	Recall	Precision	F-measure
Best individual classifier	73.10	76.76	74.76
Baseline-1	71.03	75.76	73.32
Baseline-II	71.42	75.90	73.59
Baseline-III	71.72	76.25	73.92
SOO based ensemble	74.17	77.87	75.97

- Baseline-I: Simple majority voting of the classifiers
- Baseline-II: Weighted voting where weights are based on the overall F-measure value
- Baseline-III: Weighted voting where weights are the F-measure of the individual classes

Issues of corpus compatibilities

Issues of Cross-corpus Compatibilities

- *No unified annotation scheme exists for biomedical entity annotation !!!*
- Building a system that performs reasonably well across the domains is important!
- Datasets used in the experiments
  - JNLPBA-2004 shared task
  - GENETAG
  - AIMed
- Differ in *text selection* as well as *annotation*

Experimental Setups

- Experimental Setup-I:
  - GENIA corpus by replacing all tags except ‘Protein’ by ‘O’ (other-than-NE) + AIMed corpus
  - Cross-validation

- Experimental Setup-II:
  - ‘Protein’ and ’DNA’ annotations of GENIA+ Replace all other annotations by ‘O’+ AIMed corpus
  - Cross-validation

Experiments

- Experimental Setup-III:
  - GENIA corpus by replacing all tags except 'Protein' by 'O' (other-than-NE) + GENETAG corpus
  - Test on GENETAG

- Experimental Setup-IV:
  - GENIA with only 'Protein', 'DNA' and 'RNA' annotations + GENETAG corpus
  - Test on GENETAG corpus

Results: Cross Corpus

Approach	Training set	Test set	Recall	Precision	F-measure
Best Ind. Classifier	JNLPBA (protein only)+AIMed	AIMed	83.14	83.19	83.17
SOO	JNLPBA (protein only)+AIMed	AIMed	85.10	85.01	85.05
Best Ind. Classifier	JNLPBA (protein + DNA)+AIMed	AIMed	82.17	84.15	83.15
SOO	JNLPBA (protein + DNA)+AIMed	Cross validation	84.07	86.01	85.03
Best Ind. Classifier	JNLPBA (protein only)+GENETAG	GENETAG	89.44	93.07	91.22
SOO	JNLPBA (protein only)+GENETAG	GENETAG	91.19	94.98	93.05
Best Ind. Classifier	JNLPBA (protein+DNA+RNA)+GENTAG	GENETAG	88.70	93.55	91.06
SOO	JNLPBA (protein+DNA+RNA)+GENTAG	GENETAG	90.09	95.16	92.56

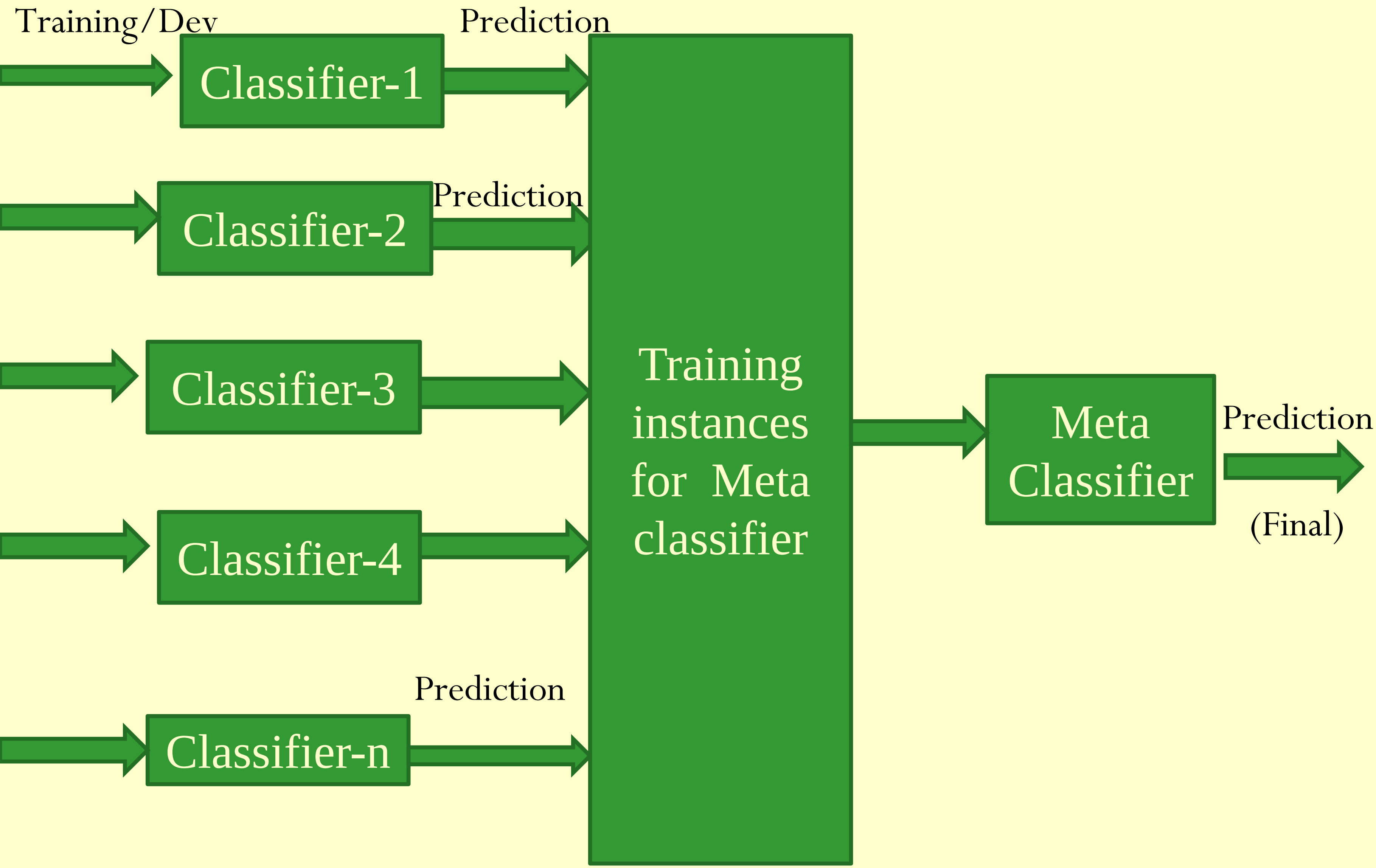
Results: Original Datasets

Dataset	Model	Recall	Precision	F-measure
GENIA	Best individual classifier	73.10	76.78	74.90
	SOO	74.17	77.87	75.97
AIMed	Best individual classifier	94.56	92.66	93.60
	SOO	95.65	94.23	94.93
GENETAG	Best individual classifier	95.35	95.31	95.33
	SOO	95.99	95.81	95.90

Drop in performance by around 10% for AIMed
and around 3% for GENETAG

Asif Ekbal and Sriparna Saha (2013). Stacked ensemble coupled with feature selection for biomedical entity extraction, ***Knowledge Based Systems****, volume (46), PP. 22–32, Elsevier.*

Stacked Model with Feature Selection



Stacked Model with Feature Selection

- Feature selection
  - GA based
  - Build few promising classifiers from the final population
  - Term them as ***base classifiers*** (CRF and SVM)
- Train the base classifiers
- Evaluate on the development data
- Meta-level training instances
  - Predictions obtained on the development data
  - Original attributes

Stacked Model with Feature Selection

- For the test set
  - Generate predictions from the base classifiers
  - Use these predictions along with the original attributes as features
- Meta classifier- CRF

Experiments (JNLPBA-2004)

Model	Recall	Precision	F-measure
Best individual classifier	73.10	76.78	74.90
Majority ensemble	71.03	75.76	73.32
Weighted ensemble	71.42	75.90	73.59
Stacked ensemble	75.15	75.20	75.17

At par the state-of-the-art system

Experiments (GENETAG)

Model	Recall	Precision	F-measure
Best individual classifier	94.41	93.50	93.95
Majority ensemble	94.45	93.65	94.05
Weighted ensemble	94.67	93.91	94.29
Stacked ensemble	95.12	94.29	94.70

At par the state-of-the-art system

NER in some specific areas

Patient Data De-identification

Problem Definition

Date

Patient Name

Hospital Name

Physician Name

Admission Date :
06/07/1999
Report Status :
Signed
Discharge Date :
06/13/1999
HISTORY OF PRESENT ILLNESS :
Essentially , Mr. Cornea is a 60 year old male who noted the onset of dark urine during early January .
He underwent CT and ERCP at the Lisonatemi Faylandsburgnic Community Hospital with a stent placement and resolution of jaundice .
He underwent an ECHO and endoscopy at Ingree and Ot of Weamanshy Medical Center on April 28 .
He was found to have a large , bulging , extrinsic mass in the lesser curvature of his stomach .
Fine needle aspiration showed atypical cells , positively reactive mesothelial cells .
Abdominal CT on April 14 , showed a 12 x 8 x 8 cm mass in the region of the left liver , and appeared to be from the lesser curvature
He denied any nausea , vomiting , anorexia , or weight loss .
He states that his color in urine or in stool is now normal .
PAST MEDICAL HISTORY :
He has hypertension and nephrolithiasis .
PAST SURGICAL HISTORY :
Status post left kidney stones x2 , and he has had a parathyroid surgery .
ALLERGIES :
He has no known drug allergies .
MEDICATIONS PRIOR TO ADMISSION :
Hydrochlorothiazide 25 mg q.d. , Clonidine 0.1 mg p.o. q.d. , baclofen 5 mg p.o. t.i.d
HOSPITAL COURSE :
Basically , patient underwent a subtotal gastrectomy on the 7th of June by Dr. Kotefooksshuff .
He had an uncomplicated postoperative course and he was transferred .
Advanced his diet on postop day # 4 to a transitional diet .
His PCA was discontinued on postop day # 4 , and essentially he was started on his pre-op medications on the postop day # 5 .
PHYSICAL EXAMINATION :

INPUT

Electronic Medical Record

OUTPUT

Electronic Medical

HISTORY OF PRESENT ILLNESS :

Mr. <PHI TYPE="PATIENT">Blind</PHI> is a 79-year-old white white male with a history of diabetes mellitus , inferior myocardial infarction , who underwent open repair of his increased diverticulum <PHI TYPE="DATE">November 13th</PHI> at <PHI TYPE="HOSPITAL">Sephsandpot Center</PHI> .

The patient developed hematemesis <PHI TYPE="DATE">November 15th</PHI> and was intubated for respiratory distress .

He was transferred to the <PHI TYPE="HOSPITAL">Valtawnprinceel Community Memorial Hospital</PHI> for endoscopy and esophagoscopy on the <PHI TYPE="DATE">16th of November</PHI> .

HISTORY OF PRESENT ILLNESS :

Mr. <XXX_PATIENT> is a 79-year-old white white male with a history of diabetes mellitus , inferior myocardial infarction , who underwent open repair of his increased diverticulum <XXX_DATE> at <XXX_HOSPITAL> .

The patient developed hematemesis <XXX_DATE> and was intubated for respiratory distress .

He was transferred to the <XXX_HOSPITAL> for endoscopy and esophagoscopy on the <XXX_DATE> .

Why De-identify Health Information?

- **Restriction of using medical records of any patient**
- Medical records have **sufficient number of personal health information (PHI)**
- Privacy does not allow to reveal all the health related information of any patient
- **Encryption of PHI terms**, according to Health Insurance Portability and Accountability Act **(HIPAA), 1996**
- Privacy Rule permits de-identification of PHI so that such information may be used and disclosed freely, without being subject to the Privacy Rule's violation

Challenges

- **Inter PHI ambiguity:** PHI terms overlap with the non-PHI terms
  E.g. **Brown (Doctor name) vs. brown (non-PHI)**
- **Intra PHI ambiguity:** One candidate word seems to belong to two or many different PHI types
  E.g. **August (Patient name) vs. August (Date)**
- **Lexical Variation:** For example, variation of the entities such as the **'50 yo m', '50 yo M', '55 YO MALE'**
- **Terminological variation and irregularities:** For example, **'3041023MARY'**
  Combination of two different PHI categories: **'3041023'** **(**represents the **MEDICALRECORD)** and **'MARY'** **(**another PHI category)

***Problem Description and Datasets**: 2014 I2b2 challenge (Stubbs et al., 2015) obtained from "Research Patient Data Repository of Partners Healthcare*

Proposed Architecture

- Basline Model: CRF based
- Deep Learning Models: RNN
  - ✓ Elman type RNN
  - ✓ Jordan type RNN

Supervised Machine Learning (CRF)

- **Context word feature** within the window of [-3,3]
- **Bag-of-word (BoW) feature:**uni-grams, bi-grams, tri-grams of the target token within the window of [-2, 2]
- **Part-of-Speech (PoS)** information within the window of [-2,2]
- **Chunk information** information within the window of [-2,2]
- **Combined PoS-token and Chunk-token Feature**

–{ W_0 POS_{-1} CH_{-1}, W_0 POS_0 CH_0, W_0 POS_{+1} CH_{+1}}

W_0 denotes the current word

POS_0 denotes the PoS of current word

CH_0 denotes the chunk information of current word

RNN: Elman-type RNN

- Every **state have the information of its previous hidden layer states** through its recurrent connections
- Hidden layer h(t) at the time instance t have the

$$h^{(1)}(t) = f(W^{(1)}C_m(x_{t-m}^{t+m}) + V^{(1)}h^{(1)}(t-1) + b)$$

- $$h^{(H)}(t) = f(W^{(H)}h^{(H-1)}(t) + V^{(H)}h^{(H)}(t-1) + b)$$

- W denote the weight connections from input layer to the hidden layer
- V denote the weight connections from hidden layer of last state to current hidden layer

RNN: Jordan-type RNN

- Inputs to the recurrent connections are through the output posterior probabilities:

$$h(t) = f(\mathbf{W}C_m(w_{t-m}^{t+m}) + \mathbf{U}P(y(t-1)) + \mathbf{b})$$

- W denote the weight connection between input to hidden layer
- U denote the weight connection between output layer of previous state to current hidden layer
- $P(y(t-1))$ is the posterior probability of last word of interest

Dataset

2014 I2b2 challenge (Stubbs et al., 2015) obtained from "Research Patient Data Repository of Partners Healthcare"

PHI Category	Train	Validation	Test
DOCTOR	2262	183	236
PATIENT	707	28	59
HOSPITAL	1342	141	164
DATE	4154	377	498
LOCATION	93	14	19
PHONE	153	12	13
ID	3200	233	264

Word Embedding

- Encoding of word into real valued vector by word2vec

Three Strategies:

1. **Random Number Initialization:** Randomly initialize the vector dimension 100 in the range −0.25 to +0.25
2. **RNN based Word Embedding:** Generated word embedding of dimension 80 trained on broadcast news corpus using RNNLM toolkit [1]
3. **Continuous bag-of-words (CBOW):** Generated word embedding of dimension 300 trained on news data corpus [1]

- **[1] T. Mikolov, http://www.fit.vutbr.cz/~imikolov/rnnlm/**

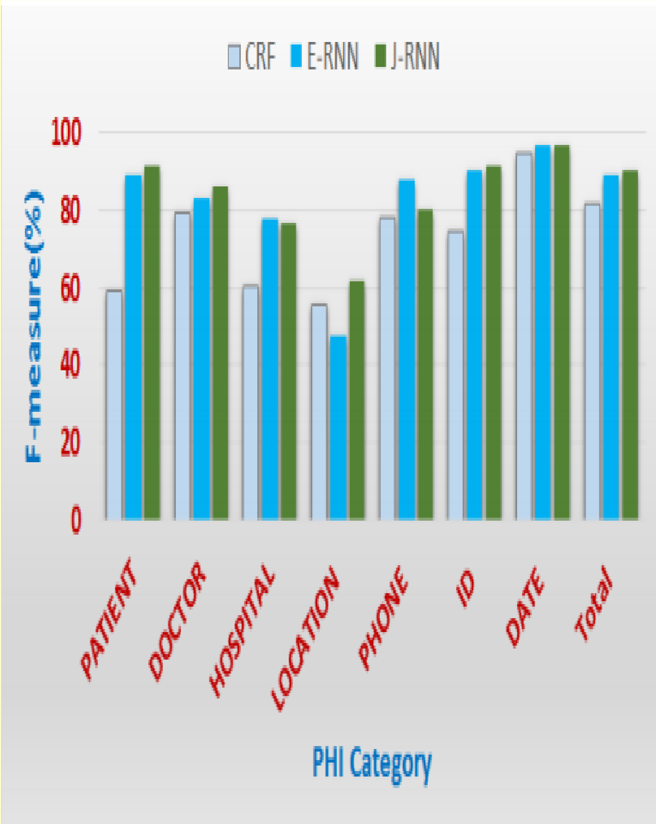
Impact of Word Embedding

Word Embedding Techniques	Dimension	Precision	Recall	F-Score
Random Number	100	87.19	85.48	86.32
RNNLM	80	88.21	87.32	87.76
CBOW	300	89.35	89.55	89.44

- **Observations:**
- **RNNLM**: **effective in capturing syntactic part** because of its **direct connection to the non-linear hidden layer**
- **CBOW:** Performs **better than RNNLM in identifying syntactic part** and **comparable on the semantic part as CBOW follow the distributional hypothesis while training**

Results : 10-fold Cross-validation

PHI Category	CRF Baseline	Elman RNN	Jordan RNN
PATIENT	58.95	88.89	91.30
DOCTOR	79.08	83.26	85.84
HOSPITAL	60.39	78.03	76.41
LOCATION	55.56	47.83	61.90
PHONE	78.26	88.00	80.00
ID	74.44	90.31	91.68
DATE	94.69	96.74	96.83
Overall	81.39	89.22	90.18



RNN Hyper-parameters

Parameter's	E-RNN	J-RNN
Hidden layer size	100	150
learning rate	0.01	0.01
Dropout probability	0.5	0.5
no. of epochs	25	25
context window size	11	9

Observations

- Two different RNN architectures **perform well over the baseline model** based on CRF
- **Jordan-RNN performs better than Elman-RNN** model for most of the **PHI category** like **Patient, Doctor, Location, ID, Date**
- **RNN model captures lexical variation** which was major source of error in CRF based model. For e.g., **"KELLIHER CARE CENTER", "KCC", "20880703"** etc
- **RNN suitable in capturing the context** due to **deeper level feature and context word** as input to model along with previous layer output
- **CRF based model** is **significantly time consuming for generating the features** for every **possible context**

Error Analysis

- **Missed Entity**
- ✓ Observed total of **106 and 95** cases in **Elman and Jordan** model, respectively
- ✓ Presence **of single-word person name with lexical variation** in case of Doctor and Patient for e.g. **"STERPSAP", "CARD"**
- ✓ **Presence of unseen terms** mostly found in **'Location', and 'Hospital'** categories for e.g. ""
- **Wrong Entity**: Total of **223 and 164** instances are mis-classified in case of **Elman and Jordan** model, respectively
- **Presence of long compounded words:** If the entity consists of more than 3 words, the system fails to identify those correctly. For example **"Tawn List Medical Center".**

Comparison RNN vs. CRF

- **NAME (Patient , Doctor, Hospital)** : RNN model was able to capture the semantic variation which was not identified by CRF based model
- **Patient** : "**KACHOLERA JUNK**", "**JUNK**"
- **Doctor**: "**Li R. Stable**", "**LI**", "**Stable**"
- **Hospital**: "**FIH**", "**KCC**", "**KELLIHER CARE CENTER**"
- **LOCATION** : RNN- Jordan model properly identifies words like **"Jer", "San"** which were **confused** with **other PHI** type in case of **CRF based model**
- **ID**: Despite of the explicit defined patterns, RNN was able to capture the token of the form **Y1WYX127C5:71** which is **difficult for CRF** to capture **without any regular expression pattern**

NER in Code-Mixed Languages
(FIRE 2016)

Joint works with Deepak Gupta, Shubham and Pushpak Bhattacharyya

Code-Mixing: Introduction

- Code-mixing refers to the mixing of two or more languages or language varieties in speech/text

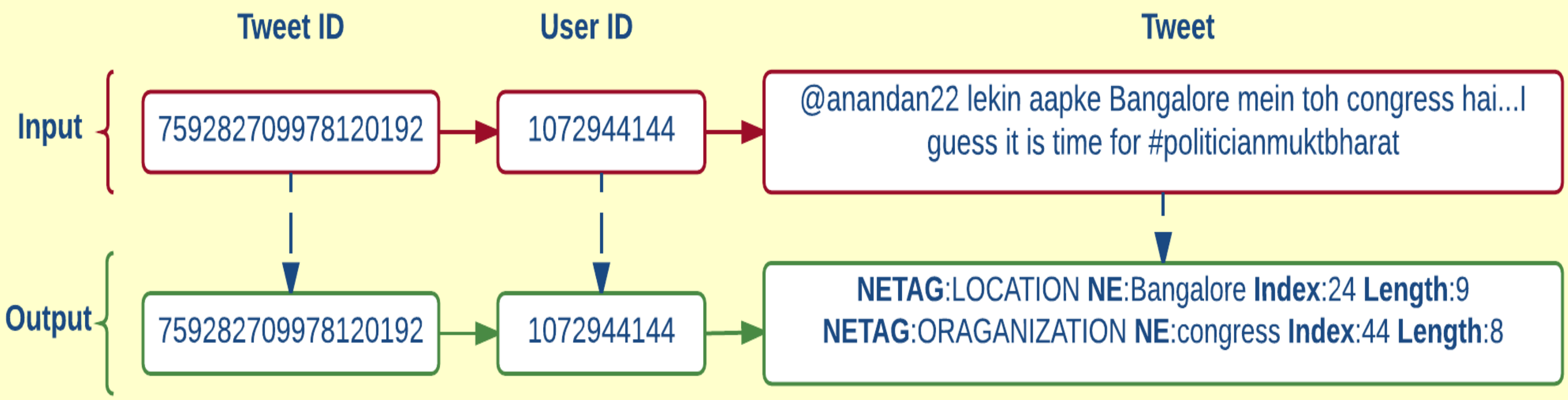
Kolkata to Varanasi ka kya distance hai

Entity **English**

Hindi

- Challenges:
  - Not limited to traditional set of named-entity classes
  - Noisy text
  - Language Identification (*a problem*!)
  - Finding effective set of features for the problem is a challenge

Overview of the Problem



Defining the problem

Let

S denotes the code-mixed sentence having n tokens $t_1, t_2, t_3 \ldots t_n$

E denotes the set of k pre-defined entities $E = \{E_1, E_2, \ldots E_k\}$

Two-Step Process:

- **Entity Extraction step**

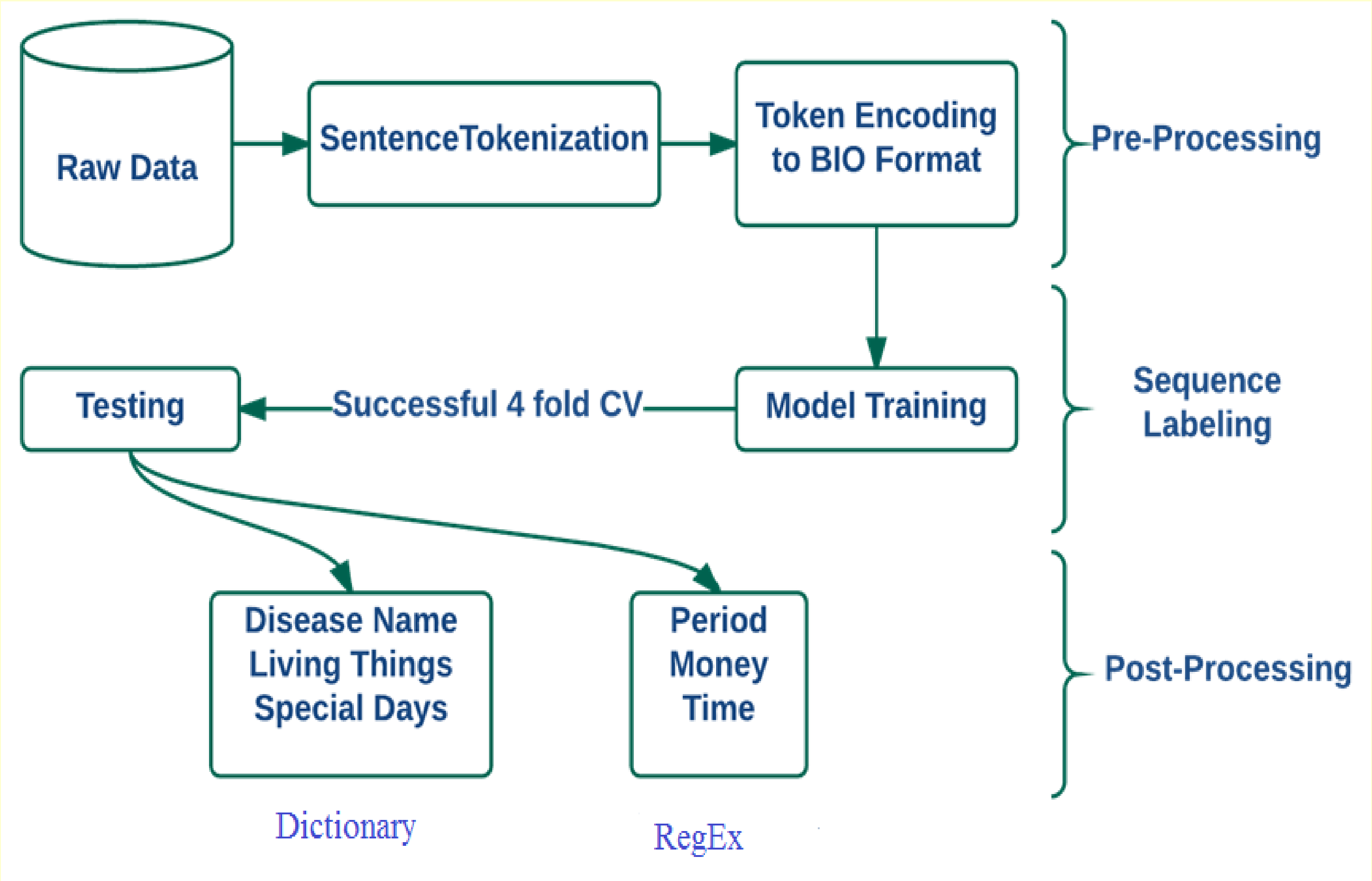
  Extract set of tokens $T_E = \{t_i, t_j \ldots t_k\}$ from S

  Denotes target NEs

- **Entity classification step**

  Classify each of the tokens of set T_E into one of the entity types

Steps of our Approach



Feature Set

@Trisolaran haha I support **Trump**. lekin don't bitch about him if y'all doung the same with afghanis @JoharJoshanda

- Word Context
- Character n-grams (1,2,3)
  - 2-gram – (t,r),(r,u),(u,m),(m,p)
  - 3-gram – (t,r,u),(r,u,m),(u,m,p)
- Word Normalization
  - Trump => Aaaaa
- Prefix/Suffix
  - Prefix = Tru, Suffix = ump
- Word Position
  - 5/18 = 0.277

Feature Set (2)

- Seen and Unseen Word Probability
  - Feature denotes the probability of a word to belong to a particular class
  - Length: Total number of output classes (*initialized with 0s*)
  - For unseen word, every bits are set to 0s
- **Two features defined for each word**
  - **Top@1 Probability:** Only the bit corresponding to the class having the highest probability is set to 1 and all the other bits set to 0
  - **Top@2 Probability:** Bit positions corresponding to the highest and second highest classes are set to 1 and others are set to 0

Feature Set (3)

- Binary-valued Features (why are these features important?)
  - **Length:** Potential entities have longer length (in this case it is 4)
  - **All Capital:** Checks whether all the characters are capitalised
  - **Init Cap:** This feature checks whether the current token starts with a capital letter or not.
  - **Init-Pun-Digit:** Checks whether the current token starts with a punctuation or a digit
  - **Digit:** Checks whether the current token contains any numeric character
  - **Hash Tag:** Checks whether current token is a hashtag (#) (why is this feature?)

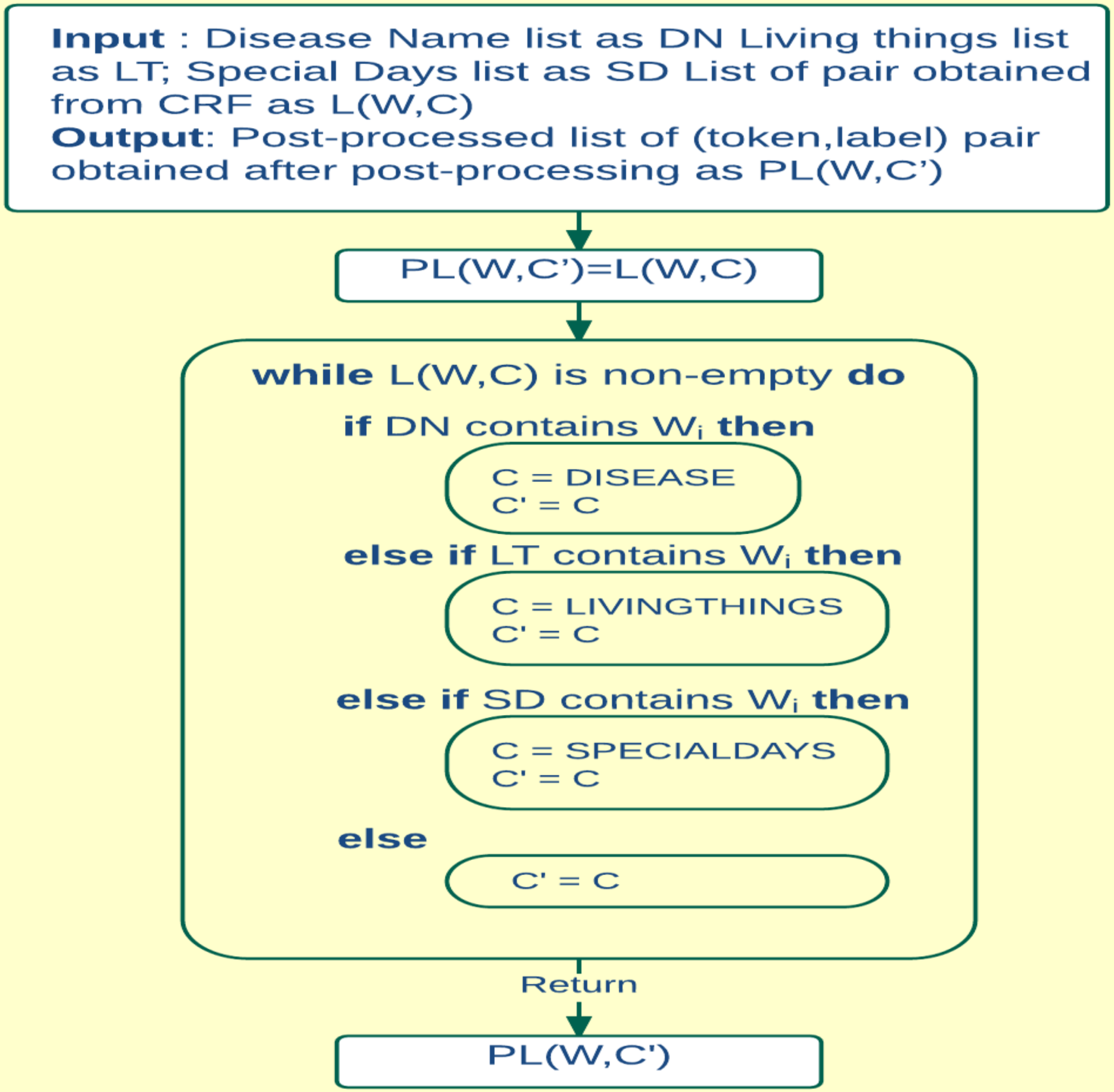
Data Set

- Domain: Tweet
- Two language pairs: **English-Hindi** and **English-Tamil** language mix
- NE types: 22
- Majority of entities are from **'Entertainment'**, **'Person'** **'Location'** and **'Organization'**
- **English-Hindi** tweet data set: Total **2700 tweets** from **2699 tweeter** users
- **English-Tamil** tweet data set: Total **2183 tweets** from **1866 tweeter** users

Data Set: Distribution

Entities	English-Hindi	English-Tamil
	# Entity	# Entity
COUNT	132	94
PLANTS	1	3
PERIOD	44	53
LOCOMOTIVE	13	5
ENTERTAINMENT	810	260
MONEY	25	66
TIME	22	18
LIVTHINGS	7	16
DISEASE	7	5
ARTIFACT	25	18
MONTH	10	25
FACILITIES	10	23
PERSON	712	661
MATERIALS	24	28
LOCATION	194	188
YEAR	143	54
DATE	33	14
ORGANIZATION	109	68
QUANTITY	2	0
DAY	67	15
SDAY	23	6
DISTANCE	0	4
Total	**2413**	**1624**

Post-Processing



Results (English-Hindi)

S. No.	Team	Run-1			Run-2			Run-3			Best-Run		
		P	R	F	P	R	F	P	R	F	P	R	F
1	Irshad-IIITHyd	80.92	59	68.24	NA			NA			80.92	59.00	68.24
2	**Deepak-IITPatna**	**81.15**	**50.39**	**62.17**	**NA**			**NA**			**81.15**	**50.39**	**62.17**
3	VeenaAmritha-T1	75.19	29.46	42.33	75	29.17	42.00	79.88	41.37	54.51	79.88	41.37	54.51
4	BharathiAmritha-T2	76.34	31.15	44.25	77.72	31.84	45.17	NA			77.72	31.84	45.17
5	Rupal-BITSPilani	58.66	32.93	42.18	58.84	35.32	44.14	59.15	34.62	43.68	58.84	35.32	44.14
6	SomnathJU	37.49	40.28	38.83	NA			NA			37.49	40.28	38.83
7	Nikhil-BITSHyd	59.28	19.64	29.50	61.8	26.39	36.99	NA			61.80	26.39	36.99
8	ShivkaranAmritha-T3	48.17	24.9	32.83	NA			NA			48.17	24.90	32.83
9	AnujSaini	72.24	18.85	29.90	NA			NA			72.24	18.85	29.90

Table : Official results obtained by the various teams participated in the CMEE-IL task- FIRE 2016 for code mixed English-Hindi language pair. Here P, R and F denotes precision, recall and F-score respectively.

Results (English-Tamil)

S. No.	Team	Run-1			Run-2			Run-3			Best-Run		
		P	R	F	P	R	F	P	R	F	P	R	F
1	**Deepak-IITPatna**	**79.92**	**30.47**	**44.12**	**NA**			**NA**			**79.92**	**30.47**	**44.12**
2	VeenaAmritha-T1	77.38	8.72	15.67	74.74	9.93	17.53	79.51	21.88	34.32	79.51	21.88	34.32
3	BharathiAmritha-T2	77.7	15.43	25.75	79.56	19.59	31.44	NA			79.56	19.59	31.44
4	RupalBITSPilani-R2	58.66	10.87	18.20	58.71	12.21	20.22	58.94	11.94	19.86	58.71	12.21	20.22
5	ShivkaranAmritha-T3	47.62	13.42	20.94	NA			NA			47.62	13.42	20.94

Table : Official results obtained by the various teams participated in the CMEE-IL task- FIRE 2016 for code mixed English-Tamil language pair. Here P, R and F denotes precision, recall and F-score respectively.

Analysis

- NEs from English-Tamil data set was particularly harder to predict due *to the transliterated text (means!!)*

- Highest *Precision* in both Hindi and Tamil
  – Hindi – 81.15%
  – Tamil – 79.92%

- Lower *F-score* on Tamil-English could be the due to the lack of good features for recognizing Tamil NE

  Language specific features could be useful

Twitter Named Entity Recognition

Joint work with Shad Akhtar and Utpal Sikdar

Named Entity Recognition (NER)

- Identify ***Person*** name, ***Location*** name, ***Organization*** name etc. in a text.

E.g. **Ashwin** said during the annual awards function in **Mumbai**

Person

Location

NER in Twitter

- Noisy and unstructured text
- Challenges
  - Short messages, *140* characters per tweet only
  - Grammar and Spelling mistakes
  - Short forms
    - ***2mrw***, ***tmrw*** for ***tomorrow***
  - Elongation
    - ***yeeeeeeeesss!!*** for ***yes!***

WNUT-2015: Named Entity Recognition in Twitter

- Coarse-grained NER
  - Identify named entities

“Junk food may not kill us directly ….” **-Velasquez-manof** #diet

Named entity

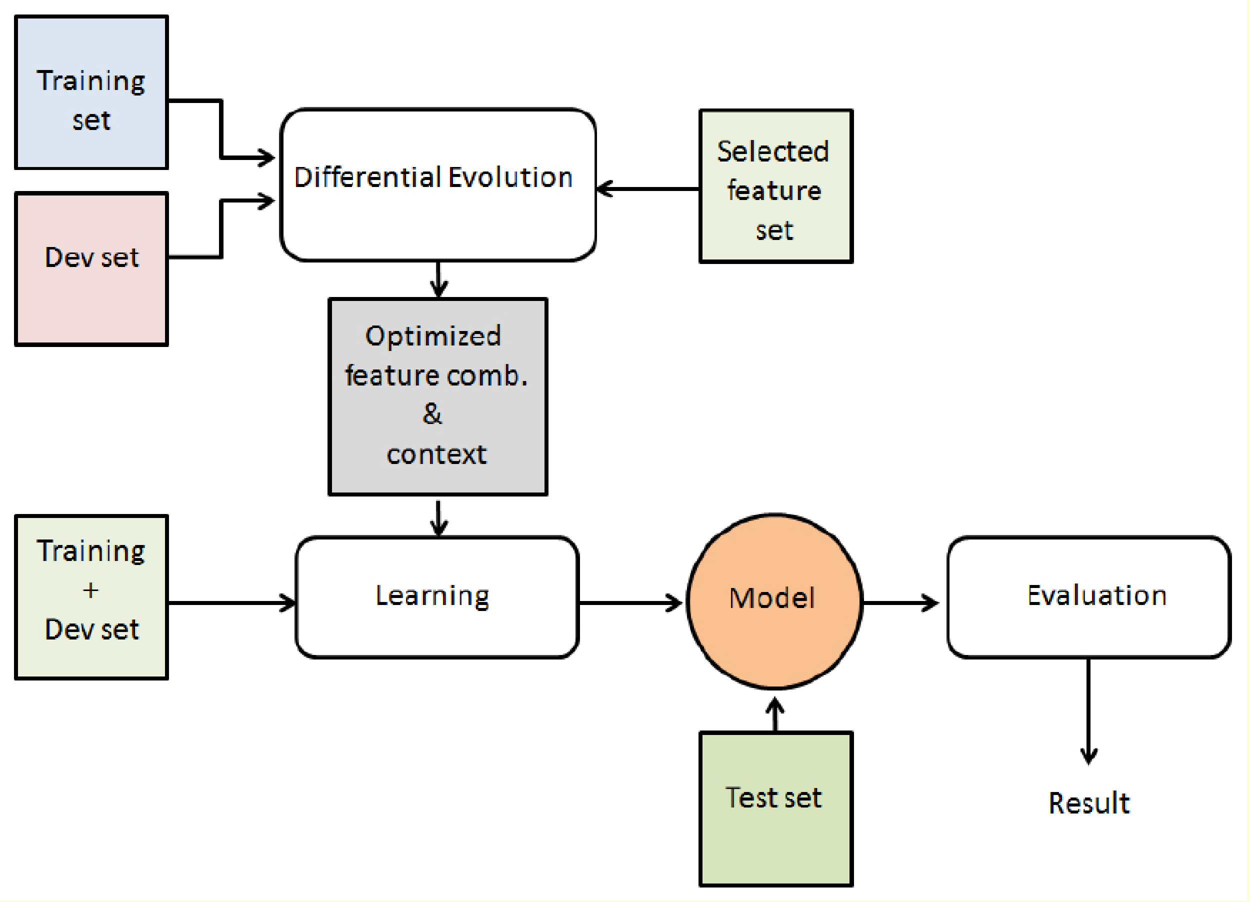
- Fine-grained NER
  - Identify named entities and their corresponding types
  - 10 types (*person, location, product, company, movie, music-artist, tv-shows, facilities, sports-team* and *others*)

“Junk food may not kill us directly ….” **-Velasquez-manof** #diet

Person name

Proposed Methodology

- Multiobjective Differential Evolution (DE) based feature selection for Twitter Named Entity Recognition
- Optimized two objectives:
  - Precision
  - Recall



Differential Evolution: Basic steps

- Initialization
- Fitness Computation
- Mutation
- Cross-over
- Selection
- Termination

Features

- **Local context** : Few previous and next tokens
- **POS tags** : Part-of-speech information
- **Word Length** : Most of the NEs are longer in length.
- **Affixes** : Suffixes and prefixes up-to length 4
- **Word Normalization** : Capital letter to ‘A’, small letter to ‘a’ and digit to ‘x’.
- **Previous occurrence** : Frequent words appeared before a NE.
- **Stop words** :
- **Uppercase**
  - **Initial capital** : First letter is in uppercase
  - **All capital** : All letters are in uppercase
  - **Inner capital** : One of the inner letter is in uppercase
- **Digit**
  - **All digit** : Token is a number
  - **Alpha digit** : Token contains character and digit.
- **First & Last word** : First and last token of a tweet.
- **Word Frequency** : Frequent words usually are non-NEs
- **Gazetteer** : NE list from training and development data.

Optimized features

Features	Coarse-grained	Fine-grained
POS	✓	✓
Word length	✓	✓
Affixes	✓	✓
Normalization	✓	✓
Previous occurrence		✓
Stop word		

Features	Coarse-grained	Fine-grained
Initial Capital	✓	
All Capital		
Inner capital		✓
All Digit	✓	
Alpha Digit		✓
Word frequency		
Gazetteer		✓

Dataset Statistics

Dataset	#Tweets	#Tokens	# NE
Train	1795	34899	1140
Dev	599	11570	356
Test	1000	16261	661

Results

Types	Model	Precision	Recall	F-measure	Accuracy
10-types	Baseline	35.56	29.05	31.97	93.41
	All features	42.41	30.00	35.14	94.94
	Proposed	60.68	29.65	**39.84**	94.54
no-type	Baseline	53.86	46.44	49.88	95.01
	All features	52.37	56.32	54.27	95.55
	Proposed	63.43	51.44	**56.81**	95.50

References

Asif Ekbal and Sriparna Saha (2013). Stacked ensemble coupled with feature selection for biomedical entity extraction, ***Knowledge Based Systems***, volume (46), PP. 22–32, Elsevier.

S.Saha, A. Ekbal and U. Sikdar (2013). Named Entity Recognition and Classification in Biomedical Text Using Classifier Ensemble. International Journal on Data Mining and Bioinformatics (in press).

A. Ekbal and S. Saha (2010). Classifier Ensemble Selection Using Genetic Algorithm for Named Entity Recognition. Research on Language and Computation (RLC), Vol. (8), PP. 73-99, Springer

A. Ekbal and S. Saha (2012). Multiobjective Optimization for Classifier Ensemble and Feature Selection: An Application to Named Entity Recognition. International Journal on Document Analysis and Recognition (IJDAR), Vol. 15(2), 143-166, Springer

A.Ekbal and S. Saha (2011). Weighted Vote-Based Classifier Ensemble for Named Entity Recognition: A Genetic Algorithm-Based Approach. ***ACM Transactions on Asian Language Information Processing (ACM TALIP***), Vol. 2(9), ACM, DOI = 10.1145/1967293.1967296 http://doi.acm.org/10.1145/1967293.1967296.