Hidden Markov Model and Graphical Models

Jie Tang
Lecture for Knowledge Engineering
Department of Computer Science and Technology
Tsinghua University
When you follow a friend in Twitter, how likely he will follow back?
Retweet Predicting

When you post a tweet...

Who will retweet it?
Binary Classifier

- Class +1
- Class -1
Sequence Labeling

• Pos Tagging
  – E.g. [He/PRP] [reckons/VBZ] [the/DT] [current/JJ] [account/NN] [deficit/NN] [will/MD] [narrow/VB] [to/TO] [only/RB] [#/#] [1.8/CD] [billion/CD] [in/IN] [September/NNP] [./.]

• Term Extraction
  – Rockwell International Corp.’s Tulsa unit said it signed a tentative agreement extending its contract with Boeing Co. to provide structural parts for Boeing’s 747 jetliners.
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...
Binary Classifier vs. Sequence Labeling

• Case restoration
  – “jack utilize outlook express to retrieve emails”
  – E.g. SVMs vs. CRFs
Sequence Labeling Problem

- Green nodes are *states*
- Purple nodes are *observations*
Example: POS Tagging Problem

Verb - Verb - Verb - Article - Verb
Noun - Noun - Preposition - Noun

Time - flies - like - an - arrow
Example: POS Tagging Problem

Time  flies  like  an  arrow
Sequence Labeling Models

- **HMM**
  - Generative model
  - E.g. Ghahramani (1997), Manning and Schutze (1999)
- **MEMM**
  - Conditional model
- **CRFs**
  - Conditional model without label bias problem
  - Linear-Chain CRFs
  - Non-Linear Chain CRFs
    - Modeling more complex interaction between labels: DCRFs, 2D-CRFs, TCRFs
General Framework

\[(O_1, S_1)\]
\[(O_2, S_2)\]
\[\ldots\]
\[(O_n, S_n)\]

Training Data

Learning System

\[P(O \mid S)\] or \[P(S \mid O)\]

Model

Test Data

Extraction System

\[(O_{n+1}, S_{n+1})\]
# Generative vs. Discriminative

<table>
<thead>
<tr>
<th>Generative</th>
<th>Discriminative</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(O \mid S)$</td>
<td>$P(S \mid O)$</td>
</tr>
<tr>
<td>Example: HMM</td>
<td>Example: MaxEnt, MEMM, CRF</td>
</tr>
<tr>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /> States generates observations</td>
<td><img src="https://via.placeholder.com/150" alt="Diagram" /> Observations (features) determine states</td>
</tr>
<tr>
<td>Learning = finding model generating observation sequence from state sequence</td>
<td>Learning = finding model mapping observation sequence to state sequence</td>
</tr>
<tr>
<td>Tagging = finding most likely state sequence having generated given observation sequence</td>
<td>Tagging = finding most likely state sequence mapped from given observation sequence</td>
</tr>
</tbody>
</table>
Assumption 1: Generative Locally Dependent Model

Hidden Markov Model (HMM)
Assumption 2: Discriminative Independent Model

Classifier:
- Maximum Entropy Model (ME)
- Support Vector Machines (SVM)
Assumption 3: Discriminative Locally Dependent Model

Maximum Entropy Markov Model (MEMM)
Assumption 4: Discriminative Globally Dependent Model

Conditional Random Field (CRF)
HMM
What is HMM?

- Green nodes are ‘*hidden*’ states
- State depends only on previous state
What is HMM?

- Purple nodes are *observations*
- Each state generates an observation
HMM Formalism

- \( s : \{1, 2, \ldots, N\} \) are values of hidden states
- \( o : \{1, 2, \ldots, M\} \) are values of observations

\[
P(O \mid S) = P(s_1)P(o_1 \mid s_1) \prod_{t=2}^{T} P(s_t \mid s_{t-1})P(o_t \mid s_t)
\]
HMM Formalism

\[ P(s \mid s') \]

\[ P(o \mid s) \]
Tagging

- Viterbi algorithm
  - given observation sequence, compute most likely having generated state sequence

$$\arg \max_S P(S \mid O) = \arg \max_S P(S, O)$$
Summary of HMM

Model
• Baum, 1966; Manning, 1999

Applications
• POS tagging (Kupiec, 1992)
• Shallow parsing (Molina, 2002; Ferran Pla, 2000; Zhou, 2000)
• Speech recognition (Rabiner, 1989; Rabiner 1993)
• Gene sequence analysis (Durbin, 1998)
• …

Limitation
• Joint probability distribution \( p(x, s) \).
• Cannot represent overlapping features and long range dependences between observed elements.
MEMM
What is MEMM?

- Green nodes are *states*
- State depends only on previous state
What is MEMM?

- Purple nodes are *observations*
- Observations (features of observations) determine states
MEMM Formalism

- \( s : \{1, 2, \ldots, N\} \) are values for states
- \( o : \{1, 2, \ldots, M\} \) are values for observations

\[
P(S \mid O) = P(s_1 \mid O) \prod_{t=2}^{T} P(s_t \mid s_{t-1}, O)
\]
MEMM Formalism

\[ s \leftarrow y, s' \leftarrow y', O \leftarrow x \]

\[ P(s \mid s', O) = P(y \mid y', x) \]

\[ P(y \mid y', x) = \frac{\exp\left(\sum_k \lambda_k f_k(x, y', y)\right)}{Z(y', x)} \]

\[ Z(y', x) = \sum_y \exp\left(\sum_k \lambda_k f_k(x, y', y)\right) \]
Inference in MEMM

• Tagging: given observation sequence, find most likely corresponding state sequence
• Learning: given observation sequence and corresponding state sequence, find model that best explains the matching
Tagging

- Viterbi algorithm

\[
\arg\max_S P(S \mid O) = \arg\max_S P(s_1 \mid O) \prod_{t=1}^{T} P(s_t \mid s_{t-1}, O)
\]
Learning

\[(x_1, y_1', y_1), (x_2, y_2', y_2), \cdots, (x_n, y_n', y_n)\]

\[P(y | y', x) = \frac{\exp(\sum_k \lambda_k f_k (x, y', y))}{Z(x, y')}\]

\[Z(x, y') = \sum_y \exp(\sum_k \lambda_k f_k (x, y', y))\]

\[\arg\max \sum_{i=1}^n \log P(y_i | y_i', x_i)\]
Learning Algorithm: IIS

**Algorithm 1**: Improved Iterative Scaling

- **Input**: Feature functions $f_1, f_2, \ldots, f_n$; empirical distribution $\tilde{p}(x, y)$
- **Output**: Optimal parameter values $\lambda^{*}_{i}$; optimal model $p_{\lambda^{*}}$

1. Start with $\lambda_i = 0$ for all $i \in \{1, 2, \ldots, n\}$

2. Do for each $i \in \{1, 2, \ldots, n\}$:
   a. Let $\Delta \lambda_i$ be the solution to
      \[
      \sum_{x, y} \tilde{p}(x)p(y|x)f_i(x, y)\exp(\Delta \lambda_i f^\#(x, y)) = \tilde{p}(f_i)
      \]
      where $f^\#(x, y) \equiv \sum_{i=1}^{n} f_i(x, y)$
      \[
      (16)
      \]
   b. Update the value of $\lambda_i$ according to: $\lambda_i \leftarrow \lambda_i + \Delta \lambda_i$

3. Go to step 2 if not all the $\lambda_i$ have converged
Summary of MEMM

• Discriminative model
• Conditional assumption
• Accuracy is higher than MaxEnt, lower than CRF
• Problem: local model $\rightarrow$ label bias problem
• MEMM contains MaxEnt as special case
Label Bias Problem

The finite-state acceptor is designed to shallow parse the sentences (chunk/phrase parsing)
1) the robot wheels Fred round
2) the robot wheels are round

Decoding it by:

\[ p(s \mid x) = p(s_1 \mid x_1) \prod_{i=2}^{n} p(s_i \mid s_{i-1}, x_i) \]

Assuming the probabilities of each of the transitions out of state 2 are approximately equal, the label bias problem means that the probability of each of these chunk sequences given an observation sequence \( x \) will also be roughly equal irrespective of the observation sequence \( x \).

On the other hand, had one of the transitions out of state 2 occurred more frequently in the training data, the probability of that transition would always be greater. This situation would result in the sequence of chunk tags associated with that path being preferred irrespective of the observation sentence.
Summary of MEMM

Model
• Berger, 1996; Ratnaparkhi 1997, 1998

Applications
• Segmentation (McCallum, 2000)
• ...

Limitation
• Label bias problem (HMM do not suffer from the label bias problem)
Conditional Markov Models (CMMs) aka MEMMs aka Maxent Taggers vs HMMS

\[ \Pr(s, o) = \prod_i \Pr(s_i | s_{i-1}) \Pr(o_i | s_{i-1}) \]

\[ \Pr(s | o) = \prod_i \Pr(s_i | s_{i-1}, o_{i-1}) \]
CRFs
MEMM to CRFs

\[ \Pr(y_1...y_n \mid x_1...x_n) = \prod_j \Pr(y_j \mid y_{j-1}, x_j) = \prod_j \frac{\exp(\sum_i \lambda_i f_i(x_j, y_j, y_{j-1}))}{Z(x_j)} \]

\[ \exp(\sum_i \lambda_i F_i(x, y)) \]

= \frac{\prod_r \prod_j Z(x_j)}{\prod_j Z(x_j)}, \text{ where } F_i(x, y) = \sum_j f_i(x_j, y_j, y_{j-1})

New model

\[ \frac{\exp(\sum_i \lambda_i F_i(x, y))}{Z(x)} \]
What is CRF?

- Green nodes are *states*
- State depends on *neighboring* states
What is CRF?

- Purple nodes are *observations*
- Observations (features of observations) determine states
CRF Formalism

- $s: \{1, 2, \ldots, N\}$ are values of states
- $o: \{1, 2, \ldots, M\}$ are values of observations

$$P(S \mid O)$$
Given an undirected graph $G = (V, E)$ such that $Y = \{Y_v | v \in V\}$, if the probability of $Y_v$ given $X$ and those random variables corresponding to nodes neighboring $v$ in $G$. Then $(X, Y)$ is a conditional random field.

undirected graphical model globally conditioned on $X$

$$p(Y_v | X, Y_u, u \neq v, \{u, v\} \in V) \iff p(Y_v | X, Y_u, (u, v) \in E)$$
Definition

CRF is a Markov Random Fields. By the Hammersley-Clifford theorem, the probability of a label can be expressed as a Gibbs distribution, so that

$$p(y | x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_j \lambda_j F_j (y, x))$$

$$F_j (y, x) = \sum_{i=1}^{n} f_j (y_{i-2}, x, i)$$

What is clique?

By only taking consideration of the one node and two nodes cliques, we have

$$p(y | x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_j \lambda_j t_j (y_{i-2}, x, i) + \sum_{k} \mu_k s_k (y_{i-2}, x, i))$$
Moreover, let us consider the problem in a first-order chain model, we have

\[ p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp \left( \sum_j \lambda_j t_j (y_{i-1}, y_i, x, i) + \sum_k \mu_k s_k (y_i, x, i) \right) \]

For simplifying description, let \( f_j(y, x) \) denote \( t_j(y_{i-1}, y_i, x, i) \) and \( s_k(y_i, x, i) \)

\[ p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp \left( \sum_j \lambda_j F_j (y, x) \right) \]

\[ F_j (y, x) = \sum_{i=1}^{\bar{n}} f_j (y_{i_{\bar{c}}}, x, i) \]
In Labeling

• In labeling, the task is to find the label sequence that has the largest probability

\[ \hat{y} = \arg \max_y p_\lambda(y \mid x) = \arg \max_y (\lambda \cdot F(y, x)) \]

\[ p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_j \lambda_j F_j(y, x)) \]

• Then the key is to estimate the parameter lambda

• Let us first review the optimization formalization
Optimization

• Defining a loss function, that should be convex for avoiding local optimization
• Defining constraints
• Finding a optimization method to solve the loss function
• A formal expression for optimization problem

$$\min_{\theta} f(x)$$

s.t.  \[ g_i(x) \geq 0, 0 \leq i \leq k \]
\[ h_j(x) = 0, 0 \leq j \leq l \]
Loss Function

Empirical loss vs. structural loss

\[ L = \sum_k |y - f(x, \lambda)| \]

min \( L \)

\[ L = \|\lambda\| + \sum_k |y - f(x, \lambda)| \]

min \( L \)

Loss function: Log-likelihood

\[ p(y \mid x, \lambda, \mu) = \frac{1}{Z} \exp(\sum_j \lambda_j F_j(y, x)) \]

\[ L(\lambda) = \sum_k \left[ -\log Z + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right] \]

\[ L = \sum_k \left[ \lambda \cdot F(y^{(k)}, x^{(k)}) - \log Z(x^{(k)}) \right] - \frac{\|\lambda\|^2}{2\sigma^2} + \text{const} \]
IIS Algorithm

First-order numerical optimization

Using Iterative Scaling (GIS, IIS)

• Initialize each $\lambda_j (= 0$ for example)
• Until convergence
  - Solve $\frac{\delta L}{\delta \lambda_j} = 0$ for each parameter $\lambda_j$
  - Update each parameter using $\lambda_j \leftarrow \lambda_j + \Delta \lambda_j$
Parameter estimation

Log-likelihood

\[ L(\lambda) = \sum_k \left[ -\log Z + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right] \]

Differentiating the log-likelihood with respect to parameter \( \lambda_j \)

\[ \frac{\delta L}{\delta \lambda_j} = \sum_k \left[ \frac{F_j(y^{(k)}, x^{(k)}) - (Z(x^{(k)}))'}{Z(x^{(k)})} \right] \]

By adding the model penalty, it can be rewritten as

\[ \frac{\delta L}{\delta \lambda_j} = \sum_k \left[ \frac{F_j(y^{(k)}, x^{(k)}) - (Z(x^{(k)}))'}{Z(x^{(k)})} \right] = \sum_y \left( \frac{\exp(\lambda \cdot F(y, x^{(k)})) \cdot F_j(y, x^{(k)})}{\sum_y \exp(\lambda \cdot F(y, x^{(k)}))} \right) - \frac{\lambda}{\sigma^2} \]
Solve the Optimization

\[ L(\lambda) = \sum_{k} \left[ -\log Z + \sum_{j} \lambda_{j} F_{j}(y^{(k)}, x^{(k)}) \right] \]

\[
\frac{\delta L}{\delta \lambda_{j}} = E_{p(y,x)}[F_{j}(Y, X)] - \sum_{k} E_{p(y|x^{(k)}, \lambda)}[F_{j}(Y, x^{(k)})]
\]

- \( E_{p(y,x)}F_{j}(y,x) \) can be calculated easily
- \( E_{p(y|x)}F_{j}(y,x) \) can be calculated by making use of a forward-backward algorithm
- \( Z \) can be estimated in the forward-backward algorithm
Forward Backward Algorithm

- An efficient algorithm using dynamic programming.

\[ \alpha_t(i) = P(o_1...o_t, s_t = i \mid \lambda) \]

\[ \beta_t(i) = P(o_t...o_T \mid s_t = i, \lambda) \]
Forward Probability

\[ \alpha_t(i) = P(o_1...o_t, s_t = i \mid \mu) \]

\[ \alpha_{t+1}(j) = \sum_{i=1}^{N} \alpha_t(i) a_{ij} b_{jo_{t+1}} \]

\[ \alpha_1(i) = \pi_i b_{io_1} \quad \pi \text{ are initial state probabilities} \]
Forward Probability

$$\alpha_{t+1}(j)$$

$$= P(o_{1...o_{t+1}}, s_{t+1} = j)$$

$$= P(o_{1...o_{t+1}} \mid s_{t+1} = j)P(s_{t+1} = j)$$

$$= P(o_{1...o_{t}} \mid s_{t+1} = j)P(o_{t+1} \mid s_{t+1} = j)P(s_{t+1} = j)$$

$$= P(o_{1...o_{t}}, s_{t+1} = j)P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1...N} P(o_{1...o_{t}}, s_{t} = i, s_{t+1} = j)P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1...N} P(o_{1...o_{t}}, s_{t} = i)P(s_{t+1} = j \mid s_{t} = i)P(o_{t+1} \mid s_{t+1} = j)$$

$$= \sum_{i=1...N} \alpha_{t}(i)a_{ij}b_{jo_{t+1}}$$
Backward Probability

\[ \beta_t(i) = P(o_t \ldots o_T \mid s_t = i) \]

\[ \beta_{T+1}(i) = 1 \]

\[ \beta_t(i) = \sum_{j=1 \ldots N} a_{ij} b_{io_t} \beta_{t+1}(j) \]
Marginal Probability

\[ p_t(i, j) = \frac{\alpha_{t-1}(i) a_{ij} b_{jo_i} \beta_t(j)}{\sum_{k=1}^{N} \alpha_t(k) \beta_t(k)} \]

\[ p_t(i) = \frac{\alpha_t(i) \beta_t(j)}{\sum_{k=1}^{N} \alpha_t(k) \beta_t(k)} \]
Calculating the Expectation

• First we define the transition matrix of $y$ for position $x$ as

$$M_i[y_{i-1}, y_i] = \exp \lambda \cdot f(y_{i-1}, y_i, x, i)$$

$$E_{p(y|x^{(k)})} \left[ F_j(Y, x^{(k)}) \right] = \sum_y p(y|x^{(k)}) F_j(y, x)$$

$$= \sum_{i=1}^n \sum_{y_{i-1}, y_i} p(y_{i-1}, y_i | x^{(k)}) f_j(y_{i-1}, y_i, x^{(k)}) + \sum_{i=1}^n \sum_j p(y_i | x^{(k)}) f_j(y_i, x^{(k)})$$

$$p(y_{i-1}, y_i | x^{(k)}) = \frac{\alpha_{i-1} (M_i * V_i) \beta_i^T}{Z(x)}$$

$$p(y_i | x^{(k)}) = \frac{\alpha_i \beta_i^T}{Z(x)}$$

$$Z(x) = \left[ \prod_{i=1}^{n+1} M_i(x) \right] = \alpha_n \cdot 1^T$$

$$\alpha_i = \begin{cases} \alpha_i M_i & 0 < i \leq n \\ 1 & i = 0 \end{cases}$$

$$\beta_i^T = \begin{cases} M_{i+1} \beta_{i+1}^T & 1 \leq i < n \\ 1 & i = n \end{cases}$$

All state features at position $i$
IIS Algorithm

First-order numerical optimization

Using Iterative Scaling (GIS, IIS)

• Initialize each \( \lambda_j (=0 \) for example) 

• Until convergence
  - Solve \( \frac{\delta L}{\delta \lambda_j} = 0 \) for each parameter \( \lambda_j \)
  - Update each parameter using \( \lambda_j \leftarrow \lambda_j + \Delta \lambda_j \)

Low efficient!!
Second-order numerical optimization

Using newton optimization technique for the parameter estimation

\[ \lambda^{(k+1)} = \lambda^{(k)} + \left( \frac{\partial^2 L}{\partial \lambda^2} \right)^{-1} \frac{\partial L}{\partial \lambda} \]

Drawbacks: parameter value initialization
And compute the second order (i.e. hesse matrix), that is difficult

Solutions:
- Conjugate-gradient (CG) (Shewchuk, 1994)
- Limited-memory quasi-Newton (L-BFGS) (Nocedal and Wright, 1999)
- Voted Perceptron (Colloins 2002)
Summary of CRFs

Model
• Lafferty, 2001

Applications
• Efficient training (Wallach, 2003)
• Training via. Gradient Tree Boosting (Dietterich, 2004)
• Bayesian Conditional Random Fields (Qi, 2005)
• Name entity (McCallum, 2003)
• Shallow parsing (Sha, 2003)
• Table extraction (Pinto, 2003)
• Signature extraction (Kristjansson, 2004)
• Accurate Information Extraction from Research Papers (Peng, 2004)
• Object Recognition (Quattoni, 2004)
• Identify Biomedical Named Entities (Tsai, 2005)
  • …

Limitation
• Huge computational cost in parameter estimation
Applications
A Unified Tagging Approach to Text Normalization (ACL’2007)

Conghui Zhu¹, Jie Tang², Hang Li³, Hwee Tou Ng⁴, and Tiejun Zhao¹

¹Harbin Institute of Technology
²Tsinghua University
³Microsoft Research Asia
⁴National University of Singapore
Outline

• Motivation
• Related Work
• Problem Description
• A Unified Tagging Approach
• Experimental Results
• Summary
Motivation

• More and more ‘informally inputted’ text data becomes available to NLP
  – E.g., emails, newsgroups, forums, blogs, etc.

• The informal text is usually very noisy
  – 98.4% of the 5000 randomly selected emails contain noises

• Previously, text normalization is conducted in a more or less ad-hoc manner
  – E.g., heuristic rules or separated classification models
I’m thinking about buying a Pocket PC device for my wife this Christmas. The worry that I have is that she won’t be able to sync it to her Outlook Express contacts.
Outline

• Motivation
• Related Work
• Problem Description
• A Unified Tagging Approach
• Experimental Results
• Summary
Related Work – Cleaning Informal Text

- Preprocessing Noisy Texts

- NER from Informal Texts

- Signature Extraction from Informal Text
  - Carvalho and Cohen (2004)

- Email Data Cleaning
  - Tang, Li, Cao, and Tang (2005)
Related Work – Language Processing

• Sentence Boundary Detection

• Case Restoration
  – Lita and Ittycheriah (2003), Mikheev (2002)

• Spelling Error Correction

• Word Normalization
  – Sproat, et al. (1999)
Outline

• Motivation
• Related Work
• Problem Description
• A Unified Tagging Approach
• Experimental Results
• Summary
Problem Description

<table>
<thead>
<tr>
<th>Level</th>
<th>Task</th>
<th>Percentages of Noises</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paragraph</td>
<td>Extra line break deletion</td>
<td>49.53</td>
</tr>
<tr>
<td></td>
<td>Paragraph boundary detection</td>
<td></td>
</tr>
<tr>
<td>Sentence</td>
<td>Extra space deletion</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Extra punctuation mark deletion</td>
<td>15.58</td>
</tr>
<tr>
<td></td>
<td>Missing space insertion</td>
<td>1.55</td>
</tr>
<tr>
<td></td>
<td>Missing punctuation mark insertion</td>
<td>3.85</td>
</tr>
<tr>
<td></td>
<td>Misused punctuation mark correction</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>Sentence boundary detection</td>
<td></td>
</tr>
<tr>
<td>Word</td>
<td>Case restoration</td>
<td>15.04</td>
</tr>
<tr>
<td></td>
<td>Unnecessary token deletion</td>
<td>9.69</td>
</tr>
<tr>
<td></td>
<td>Misspelled word correction</td>
<td>3.41</td>
</tr>
</tbody>
</table>

An ideal normalization method should consider processing all the tasks together!

Text normalization is defined at three levels.
Outline

• Motivation
• Related Work
• Problem Description
• A Unified Tagging Approach
• Experimental Results
• Summary
Processing Flow

Preprocessing

Determine Tokens

Standard word

Non-standard word

Punc. mark

Space

Line break

Assigning tags

Paragraph segmentation

Paragraphs

Train

Test

Labeling data

Model Learning

Feature definitions

Learning a CRF model

A unified tagging model

Labeled data

Tagging results

i’m thinking about buying a pocket...
i’m also considering buying an ipaq...

get

a

get

a

toshiba’s

pc
### Token Definitions

<table>
<thead>
<tr>
<th>Standard word</th>
<th>Words in natural language</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-standard word</strong></td>
<td>Including several general ‘special words’ e.g. email address, IP address, URL, date, number, money, percentage, unnecessary tokens (e.g. ‘===’ and ‘###’), etc.</td>
</tr>
<tr>
<td><strong>Punctuation marks</strong></td>
<td>Including period, question mark, and exclamation mark</td>
</tr>
<tr>
<td><strong>Space</strong></td>
<td>Each space will be identified as a space token</td>
</tr>
<tr>
<td><strong>Line break</strong></td>
<td>Every line break is a token</td>
</tr>
</tbody>
</table>
Possible Tags Assignment

- **Green** nodes are *tags*
- **Purple** nodes are *tokens*
Tagging

\[ Y^* = \max_Y P(Y|X), \text{ where } X \text{ – tokens, } Y \text{ – tags} \]
## Features

<table>
<thead>
<tr>
<th>Transition Features</th>
<th>State Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_{i-1} = y'$, $y_i = y$</td>
<td>$w_i = w$, $y_i = y$</td>
</tr>
<tr>
<td>$y_{i-1} = y'$, $y_i = y$, $w_i = w$</td>
<td>$w_{i+2} = w$, $y_i = y$</td>
</tr>
<tr>
<td>$y_{i-1} = y'$, $y_i = y$, $t_i = t$</td>
<td>$w_{i+3} = w$, $y_i = y$</td>
</tr>
</tbody>
</table>

In total, more than 4M features were used in our experiments.
Outline

• Motivation
• Related Work
• Problem Description
• A Unified Tagging Approach
• Experimental Results
• Summary
## Datasets in Experiments

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Number of Email</th>
<th>Number of Noises</th>
<th>Extra Line Break</th>
<th>Extra Space</th>
<th>Extra Punc.</th>
<th>Missing Space</th>
<th>Missing Punc.</th>
<th>Casing Error</th>
<th>Spelling Error</th>
<th>Misused Punc.</th>
<th>Unnecessary Token</th>
<th>Number of Paragraph Boundary</th>
<th>Number of Sentence Boundary</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>100</td>
<td>702</td>
<td>476</td>
<td>31</td>
<td>8</td>
<td>3</td>
<td>24</td>
<td>53</td>
<td>14</td>
<td>2</td>
<td>91</td>
<td>457</td>
<td>291</td>
</tr>
<tr>
<td>Ontology</td>
<td>100</td>
<td>2,731</td>
<td>2,132</td>
<td>24</td>
<td>3</td>
<td>10</td>
<td>68</td>
<td>205</td>
<td>79</td>
<td>15</td>
<td>195</td>
<td>677</td>
<td>1,132</td>
</tr>
<tr>
<td>NLP</td>
<td>60</td>
<td>861</td>
<td>623</td>
<td>12</td>
<td>1</td>
<td>3</td>
<td>23</td>
<td>135</td>
<td>13</td>
<td>2</td>
<td>49</td>
<td>244</td>
<td>296</td>
</tr>
<tr>
<td>ML</td>
<td>40</td>
<td>980</td>
<td>868</td>
<td>17</td>
<td>0</td>
<td>2</td>
<td>13</td>
<td>12</td>
<td>7</td>
<td>0</td>
<td>61</td>
<td>240</td>
<td>589</td>
</tr>
<tr>
<td>Jena</td>
<td>700</td>
<td>5,833</td>
<td>3,066</td>
<td>117</td>
<td>42</td>
<td>38</td>
<td>234</td>
<td>888</td>
<td>288</td>
<td>59</td>
<td>1,101</td>
<td>2,999</td>
<td>1,836</td>
</tr>
<tr>
<td>Weka</td>
<td>200</td>
<td>1,721</td>
<td>886</td>
<td>44</td>
<td>0</td>
<td>30</td>
<td>37</td>
<td>295</td>
<td>77</td>
<td>13</td>
<td>339</td>
<td>699</td>
<td>602</td>
</tr>
<tr>
<td>Protégé</td>
<td>700</td>
<td>3,306</td>
<td>1,770</td>
<td>127</td>
<td>48</td>
<td>151</td>
<td>136</td>
<td>552</td>
<td>116</td>
<td>9</td>
<td>397</td>
<td>1,645</td>
<td>1,035</td>
</tr>
<tr>
<td>OWL</td>
<td>300</td>
<td>1,232</td>
<td>680</td>
<td>43</td>
<td>24</td>
<td>47</td>
<td>41</td>
<td>152</td>
<td>44</td>
<td>3</td>
<td>198</td>
<td>578</td>
<td>424</td>
</tr>
<tr>
<td>Mobility</td>
<td>400</td>
<td>2,296</td>
<td>1,292</td>
<td>64</td>
<td>22</td>
<td>35</td>
<td>87</td>
<td>495</td>
<td>92</td>
<td>8</td>
<td>201</td>
<td>891</td>
<td>892</td>
</tr>
<tr>
<td>WinServer</td>
<td>400</td>
<td>3,487</td>
<td>2,029</td>
<td>59</td>
<td>26</td>
<td>57</td>
<td>142</td>
<td>822</td>
<td>121</td>
<td>21</td>
<td>210</td>
<td>1,232</td>
<td>1,151</td>
</tr>
<tr>
<td>Windows</td>
<td>1,000</td>
<td>9,293</td>
<td>3,416</td>
<td>3,056</td>
<td>60</td>
<td>116</td>
<td>348</td>
<td>1,309</td>
<td>291</td>
<td>67</td>
<td>630</td>
<td>3,581</td>
<td>2,742</td>
</tr>
<tr>
<td>PSS</td>
<td>1,000</td>
<td>8,965</td>
<td>3,348</td>
<td>2,880</td>
<td>59</td>
<td>153</td>
<td>296</td>
<td>1,331</td>
<td>276</td>
<td>66</td>
<td>556</td>
<td>3,411</td>
<td>2,590</td>
</tr>
<tr>
<td>Total</td>
<td>5,000</td>
<td>41,407</td>
<td>20,586</td>
<td>6,474</td>
<td>293</td>
<td>645</td>
<td>1,449</td>
<td>6,249</td>
<td>1,418</td>
<td>265</td>
<td>4,028</td>
<td>16,654</td>
<td>13,580</td>
</tr>
</tbody>
</table>
Baseline Methods

Two baselines: cascaded and independent methods

Cascaded

- Extra line break detection
- Extra space detection
- Extra punc. mark detection
- Sentence boundary detection
- Unnecessary token deletion
- Case restoration

Independent

- Extra line break detection
- Extra space detection
- Extra punc. mark detection
- Sentence boundary detection
- Unnecessary token deletion
- Case restoration

Heuristic rules

SVM

TrueCasing/CRF
### Normalization Results—5-fold cross validation

<table>
<thead>
<tr>
<th>Detection Task</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F1-measure</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extra Line Break</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent</td>
<td>95.16</td>
<td>91.52</td>
<td>93.30</td>
<td>93.81</td>
</tr>
<tr>
<td>Cascaded</td>
<td>95.16</td>
<td>91.52</td>
<td>93.30</td>
<td>93.81</td>
</tr>
<tr>
<td>Unified</td>
<td>93.87</td>
<td>93.63</td>
<td><strong>93.75</strong></td>
<td><strong>94.53</strong></td>
</tr>
<tr>
<td><strong>Extra Space</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent</td>
<td>91.85</td>
<td>94.64</td>
<td>93.22</td>
<td>99.87</td>
</tr>
<tr>
<td>Cascaded</td>
<td>94.54</td>
<td>94.56</td>
<td>94.55</td>
<td>99.89</td>
</tr>
<tr>
<td>Unified</td>
<td>95.17</td>
<td>93.98</td>
<td><strong>94.57</strong></td>
<td><strong>99.90</strong></td>
</tr>
<tr>
<td><strong>Extra Punctuation Mark</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent</td>
<td>88.63</td>
<td>82.69</td>
<td>85.56</td>
<td>99.66</td>
</tr>
<tr>
<td>Cascaded</td>
<td>87.17</td>
<td>85.37</td>
<td>86.26</td>
<td>99.66</td>
</tr>
<tr>
<td>Unified</td>
<td>90.94</td>
<td>84.84</td>
<td><strong>87.78</strong></td>
<td><strong>99.71</strong></td>
</tr>
<tr>
<td><strong>Sentence Boundary</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent</td>
<td>98.46</td>
<td>99.62</td>
<td>99.04</td>
<td>98.36</td>
</tr>
<tr>
<td>Cascaded</td>
<td>98.55</td>
<td>99.20</td>
<td>98.87</td>
<td>98.08</td>
</tr>
<tr>
<td>Unified</td>
<td>98.76</td>
<td>99.61</td>
<td><strong>99.18</strong></td>
<td><strong>98.61</strong></td>
</tr>
<tr>
<td><strong>Unnecessary Token</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent</td>
<td>72.51</td>
<td>100.0</td>
<td>84.06</td>
<td>84.27</td>
</tr>
<tr>
<td>Cascaded</td>
<td>72.51</td>
<td>100.0</td>
<td>84.06</td>
<td>84.27</td>
</tr>
<tr>
<td>Unified</td>
<td>98.06</td>
<td>95.47</td>
<td><strong>96.75</strong></td>
<td><strong>96.18</strong></td>
</tr>
<tr>
<td><strong>Case Restoration (TrueCasing)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent</td>
<td>27.32</td>
<td>87.44</td>
<td>41.63</td>
<td>96.22</td>
</tr>
<tr>
<td>Cascaded</td>
<td>28.04</td>
<td>88.21</td>
<td>42.55</td>
<td>96.35</td>
</tr>
<tr>
<td><strong>Case Restoration (CRF)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Independent</td>
<td>84.96</td>
<td>62.79</td>
<td>72.21</td>
<td>99.01</td>
</tr>
<tr>
<td>Cascaded</td>
<td>85.85</td>
<td>63.99</td>
<td>73.33</td>
<td>99.07</td>
</tr>
<tr>
<td>Unified</td>
<td>86.65</td>
<td>67.09</td>
<td><strong>75.63</strong></td>
<td><strong>99.21</strong></td>
</tr>
</tbody>
</table>
Normalization Results (cont.)

<table>
<thead>
<tr>
<th>Text Normalization</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F1</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent (TrueCasing)</td>
<td>69.54</td>
<td>91.33</td>
<td>78.96</td>
<td>97.90</td>
</tr>
<tr>
<td>Independent (CRF)</td>
<td>85.05</td>
<td>92.52</td>
<td>88.63</td>
<td>98.91</td>
</tr>
<tr>
<td>Cascaded (TrueCasing)</td>
<td>70.29</td>
<td>92.07</td>
<td>79.72</td>
<td>97.88</td>
</tr>
<tr>
<td>Cascaded (CRF)</td>
<td>85.06</td>
<td>92.70</td>
<td>88.72</td>
<td>98.92</td>
</tr>
<tr>
<td>Unified w/o Transition Features</td>
<td>86.03</td>
<td>93.45</td>
<td>89.59</td>
<td>99.01</td>
</tr>
<tr>
<td>Unified</td>
<td>86.46</td>
<td>93.92</td>
<td>90.04</td>
<td>99.05</td>
</tr>
</tbody>
</table>

1) The baseline methods suffered from ignorance of the dependencies between the subtasks
2) Our method benefits from modeling the dependencies
I’m thinking about buying a **pocket PC** device for my wife this Christmas, The worry that I have is that she won’t be able to sync it to her **Outlook Express** contacts.

---

**Comparison Example**

By independent method

1. I’m thinking about buying a **pocket**
2. **PC** device for my wife this Christmas.
3. The worry that I have is that she won’t
4. be able to sync it to her **Outlook Express**
5. contacts.

By cascaded method

1. I’m thinking about buying a **pocket PC**
2. device for my wife this Christmas,
3. the worry that I have is that she won’t be able to sync it to her **Outlook Express** contacts.

By our method

I’m thinking about buying a **Pocket PC**

device for my wife this Christmas.

The worry that I have is that she won’t be able to sync it to her Outlook Express contacts.
Error Analysis

• Extra line break detection
  – 31.14% due to incorrect elimination and 64.07% due to overlooking extra line breaks

• Space detection
  – e.g. “02-16- 2006” and “desk top”

• Case restoration
  – e.g. special word “.NET” and “Ph.D.” and Proper nouns like “John” and “HP Compaq”
## Computational Cost

<table>
<thead>
<tr>
<th>Methods</th>
<th>Training</th>
<th>Tagging</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent (TrueCasing)</td>
<td>2 minutes</td>
<td>a few seconds</td>
</tr>
<tr>
<td>Cascaded (TrueCasing)</td>
<td>3 minutes</td>
<td>a few seconds</td>
</tr>
<tr>
<td>Unified</td>
<td>5 hours</td>
<td>25s</td>
</tr>
</tbody>
</table>

*Tested on a computer with two 2.8G P4-CPUs and 3G memory*
How Text Normalization Helps NER

Percentage (%)

F1-Measure

Original
Independent
Cascaded
Unified
Clean

+16.60%
Outline

• Motivation
• Related Work
• Problem Description
• A Unified Tagging Approach
• Experimental Results
• Summary
Summary

- Investigated the problem of text normalization
- Formalized the problem as a task of noise elimination and boundary detection subtasks
- Proposed a unified tagging approach to perform the subtasks together
- Empirical verification of the effectiveness of the proposed approach
Thanks!

HP: http://keg.cs.tsinghua.edu.cn/~jietang/

Email: jietang@tsinghua.edu.cn