

# CSE 842

# Natural Language Processing

## Lecture 18: Maximum Entropy Model and Information Extraction

# Classification

Given a set of classes:  $C = [c_1, c_2, \dots, c_n]$  and an observation  $x$ , the task is to identify which element from  $C$  the observation  $x$  belongs to.

- Examples:
  - Whether a pair of noun phrases corefer
  - End-of-sentence boundaries
  - Email spam recognition
  - Sentiment analysis
  - Word sense disambiguation
  - Text classification

# Supervised Learning

- Given a set of pairs  $(x, y)$  where  $y$  is a label (or class) and  $x$  is an observation, discover a function that assigns the correct labels to the  $x$ .
- Functions could be:
  - Rules
  - Decision trees
  - Probabilistic models
  - Etc.
- What we have encountered so far:
  - Decision List
  - Naïve Bayes
  - Hidden Markov Model (sequence model)

# A Probabilistic Classifier

- Predict a probability distribution over all classes for a given input pattern.
- General problem:
  - An input domain  $X$ ,
  - A finite class domain  $Y$
  - The goal is to provide a conditional probability  $P(y | x)$  for any  $x, y$  where  $x \in X$  and  $y \in Y$ .
- Today:
  - A brief introduction to Maximum Entropy Model

# A Simple Example

- Consider a translation example
- English 'in'  $\rightarrow$  French  $\{dans, en, \grave{a}, au-cours-de, pendant\}$
- Goal:  $p(dans)$ ,  $p(en)$ ,  $p(\grave{a})$ ,  $p(au-cours-de)$ ,  $p(pendant)$
- Case 1: no prior knowledge on translation
  - What is your guess of the probabilities?

# A Simple Example

- Consider a translation example
- English 'in'  $\rightarrow$  French  $\{dans, en, \grave{a}, au\ cours\ de, pendant\}$
- Goal:  $p(dans)$ ,  $p(en)$ ,  $p(\grave{a})$ ,  $p(au-cours-de)$ ,  $p(pendant)$
- Case 1: no prior knowledge on translation
  - What is your guess of the probabilities?
  - $p(dans)=p(en)=p(\grave{a})=p(au-cours-de)=p(pendant)=1/5$
- Case 2: 30% of times either *dans* or *en* is used

# A Simple Example

- Consider a translation example
- English 'in'  $\rightarrow$  French  $\{dans, en, \grave{a}, au\ cours\ de, pendant\}$
- Goal:  $p(dans)$ ,  $p(en)$ ,  $p(\grave{a})$ ,  $p(au-cours-de)$ ,  $p(pendant)$
- Case 1: no prior knowledge on translation
  - What is your guess of the probabilities?
  - $p(dans)=p(en)=p(\grave{a})=p(au-cours-de)=p(pendant)=1/5$
- Case 2: 30% of times either *dans* or *en* is used
  - What is your guess of the probabilities?
  - $p(dans)=p(en)=3/20$   $p(\grave{a})=p(au-cours-de)=p(pendant)=7/30$
- Uniform distribution is favored

# A Simple Example

- Case 3: 30% of time *dans* or *en* is used, and 50% of times *dans* or *à* is used
  - What is your guess of the probabilities?

$$p(\mathbf{dans}) + p(\mathbf{en}) = 3/10$$

$$p(\mathbf{dans}) + p(\mathbf{en}) + p(\mathbf{\grave{a}}) + p(\mathbf{au\ cours\ de}) + p(\mathbf{pendant}) = 1$$

$$p(\mathbf{dans}) + p(\mathbf{\grave{a}}) = 1/2$$

# A Simple Example

- Case 3: 30% of time *dans* or *en* is used, and 50% of times *dans* or *à* is used
  - What is your guess of the probabilities?

$$p(\mathit{dans}) + p(\mathit{en}) = 3/10$$

$$p(\mathit{dans}) + p(\mathit{en}) + p(\mathit{\grave{a}}) + p(\mathit{au\ cours\ de}) + p(\mathit{pendant}) = 1$$

$$p(\mathit{dans}) + p(\mathit{\grave{a}}) = 1/2$$

- A good probability distribution should
  - Satisfy the constraints
  - Be close to uniform distribution

# Maximum Entropy (MaxEnt)

- A uniformity of distribution is measured by entropy of the distribution

$$P^* = \max_P H(P)$$

where  $H(P) = -p(dans) \log p(dans) - p(en) \log p(en) - p(a) \log p(a) - p(\text{au - course - de}) \log p(\text{au - course - de}) - p(\text{pendant}) \log p(\text{pendant})$   
subject to

$$p(dans) + p(en) = 3/10$$

$$p(dans) + p(a) = 1/2$$

$$p(dans) + p(en) + p(a) + p(\text{au - cours - de}) + p(\text{pendant}) = 1$$

- Solution:  $p(dans) = 0.2$ ,  $p(a) = 0.3$ ,  $p(en) = 0.1$ ,  $p(\text{au-cours-de}) = 0.2$ ,  $p(\text{pendant}) = 0.2$

# Another Example

- Task: estimate joint model  $p(a,b)$  where  $b$  is a word, and  $a \in \{NNP, VBG\}$
- Evidence in the corpus:
  - 4 capitalized words with NNP (Proper Noun)
  - 10 words total

$P(a,b)$	$a=NNP$	$a=VBG$	total
b is cap	0.4	?	
b is not cap	?	?	
			1.0

# Another Example

One possible solution:  $H(p) = 1.46$

P(a,b)	a=NNP	a=VBG	total
b is cap	0.4	<b>0.05</b>	
b is not cap	<b>0.5</b>	<b>0.05</b>	
			1.0

# Another Example

MaxEnt Solution:  $H(p) = 1.92$

P(a,b)	a=NNP	a=VBG	total
b is cap	0.4	<b>0.2</b>	
b is not cap	<b>0.2</b>	<b>0.2</b>	
			1.0

Most uncertain way to satisfy constraints from the observations

# Principle of Maximum Entropy

- Use the probability distribution that has maximum entropy, or that is maximally uncertain, from those that are consistent with observed evidence
- $P = \{ \text{models consistent with evidence} \}$
- $H(p) = \text{entropy of } p$
- $P_{ME} = \operatorname{argmax}_{p \in P} H(p)$

# MaxEnt for Classification Problems

- Want a  $p(y|x)$  to be close to a uniform distribution
  - Maximize the conditional entropy of training data

$$H(y | \vec{x}) = \sum_{i=1}^N H(y | \vec{x}_i) = -\sum_{i=1}^N \sum_y p(y | \vec{x}_i) \log p(y | \vec{x}_i)$$

# MaxEnt for Classification Problems

- Want a  $p(y|x)$  to be close to a uniform distribution
  - Maximize the conditional entropy of training data

$$H(y | \vec{x}) = \sum_{i=1}^N H(y | \vec{x}_i) = - \sum_{i=1}^N \sum_{y \in Y} p(y | \vec{x}_i) \log p(y | \vec{x}_i)$$

- Constraints
  - Valid probability distribution

$$\forall i \sum_y p(y | \vec{x}_i) = 1$$

- From training data: the model should be consistent with data
  - For each class, model mean of  $\mathbf{x}$  = empirical mean of  $\mathbf{x}$

$$\forall y \in [1, 2, \dots, C] \quad \frac{1}{N} \sum_{i=1}^N p(y | \vec{x}_i) \vec{x}_i = \frac{1}{N} \sum_{i=1}^N \vec{x}_i \delta(y, y_i)$$

# MaxEnt Model

- Consistency with data is ensured by the equality constraints

$$\sum_{i=1}^N p(y | \vec{x}_i) \vec{x}_i = \sum_{i=1}^N \vec{x}_i \delta(y, y_i)$$

- For each feature, the empirical mean equals to the model mean
- Beyond feature vector  $\mathbf{x}$ :

$$\sum_{i=1}^N p(y | \vec{x}_i) f_k(\vec{x}_i) = \sum_{i=1}^N f_k(\vec{x}_i) \delta(y, y_i)$$

# Solution to MaxEnt

- It turns out the solution is just conditional exponential model without thresholds

$$p(y | \vec{x}) = \frac{\exp(\vec{w}_y \cdot \vec{x})}{\sum_y \exp(\vec{w}_y \cdot \vec{x})}$$

# MaxEnt in NLP

- Features: often indicator functions which considers both some observation property and a particular class

$$f : X \times Y \rightarrow \{0,1\}$$

$$f(x, y) = \begin{cases} 1 & \text{if } (y = c) \text{ and } q(x) \\ 0 & \text{otherwise} \end{cases}$$

$$p(y | x) = \frac{1}{Z} \exp\left(\sum_{i=1}^M w_i f_i(x, y)\right)$$

$$Z = \sum_{y' \in Y} p(y' | x) = \sum_{y' \in Y} \exp\left(\sum_{i=1}^M w_i f_i(x, y')\right)$$

# Example on POS Tagging

Hispaniola/NNP quickly/RB became/VB an/DT  
important/JJ base/?? from which Spain expanded  
its empire into the rest of the Western Hemisphere .

- There are many possible tags in the position ??  
{NN, NNS, Vt, Vi, IN, DT, . . . }

- The task: model the distribution

$$P(t_i | t_1, \dots, t_{i-1}, w_1, \dots, w_n)$$

where  $t_i$  is the  $i$ 'th tag in the sequence,  $w_i$  is the  $i$ 'th word

# Feature Representation for POS Tagging

- Each  $x$  is a “history” of the form  $\langle t_1, \dots, t_{i-1}, w_1, \dots, w_n, i \rangle$
- Each  $y$  is a POS tag, such as NN, NNS, IN, DT, ...

We have  $m$  features  $f_k(x, y)$  for  $k = 1, \dots, m$

For example:

$$f_1(x, y) = \begin{cases} 1 & \text{if current word } w_i \text{ is "base" and } y = Vt \\ 0 & \text{otherwise} \end{cases}$$

$$f_2(x, y) = \begin{cases} 1 & \text{if current word } w_i \text{ ends in "ing" and } y = VBG \\ 0 & \text{otherwise} \end{cases}$$

## The Full Set of Features in [Ratnaparkhi 96]

- Word/tag features for all word/tag pairs, e.g.,

$$\phi_{100}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ is base and } t = \text{Vt} \\ 0 & \text{otherwise} \end{cases}$$

- Spelling features for all prefixes/suffixes of length  $\leq 4$ , e.g.,

$$\phi_{101}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ ends in ing and } t = \text{VBG} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{102}(h, t) = \begin{cases} 1 & \text{if current word } w_i \text{ starts with pre and } t = \text{NN} \\ 0 & \text{otherwise} \end{cases}$$

*f(x,y) is represented by  $\phi(h,t)$  here*

## The Full Set of Features in [Ratnaparkhi 96]

- Contextual Features, e.g.,

$$\phi_{103}(h, t) = \begin{cases} 1 & \text{if } \langle t_{-2}, t_{-1}, t \rangle = \langle \text{DT}, \text{JJ}, \text{Vt} \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{104}(h, t) = \begin{cases} 1 & \text{if } \langle t_{-1}, t \rangle = \langle \text{JJ}, \text{Vt} \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{105}(h, t) = \begin{cases} 1 & \text{if } \langle t \rangle = \langle \text{Vt} \rangle \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{106}(h, t) = \begin{cases} 1 & \text{if previous word } w_{i-1} = \textit{the} \text{ and } t = \text{Vt} \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_{107}(h, t) = \begin{cases} 1 & \text{if next word } w_{i+1} = \textit{the} \text{ and } t = \text{Vt} \\ 0 & \text{otherwise} \end{cases}$$

# Feature Vector

- We can come up with practically any questions (features)
- For any input pattern  $x$ , each class is map to a different feature vector.

$$\begin{aligned} f(\langle \text{JJ, DT, } \langle \text{Hispaniola, } \dots \rangle, 6 \rangle, \text{Vt}) &= 1001011001001100110 \\ f(\langle \text{JJ, DT, } \langle \text{Hispaniola, } \dots \rangle, 6 \rangle, \text{JJ}) &= 0110010101011110010 \\ f(\langle \text{JJ, DT, } \langle \text{Hispaniola, } \dots \rangle, 6 \rangle, \text{NN}) &= 0001111101001100100 \\ f(\langle \text{JJ, DT, } \langle \text{Hispaniola, } \dots \rangle, 6 \rangle, \text{IN}) &= 0001011011000000010 \end{aligned}$$

# Learning Weights

- Maximum-likelihood estimates, given training sample  $(x_i, y_i)$  for  $i = 1..n$ , find  $w$  that maximize the log likelihood

$$L(\mathbf{w}) = \sum_{i=1}^n \log p(y_i | x_i) = \sum_{i=1}^n \mathbf{w} \cdot \mathbf{f}(x_i, y_i) - \sum_{i=1}^n \log \sum_{y' \in Y} e^{\mathbf{w} \cdot \mathbf{f}(x_i, y')}$$

- Smooth of weights by regularization

$$L(\mathbf{w}) = \sum_{i=1}^n \log p(y_i | x_i) = \sum_{i=1}^n \mathbf{w} \cdot \mathbf{f}(x_i, y_i) - \sum_{i=1}^n \log \sum_{y' \in Y} e^{\mathbf{w} \cdot \mathbf{f}(x_i, y')} - \alpha \|\mathbf{w}\|_2$$

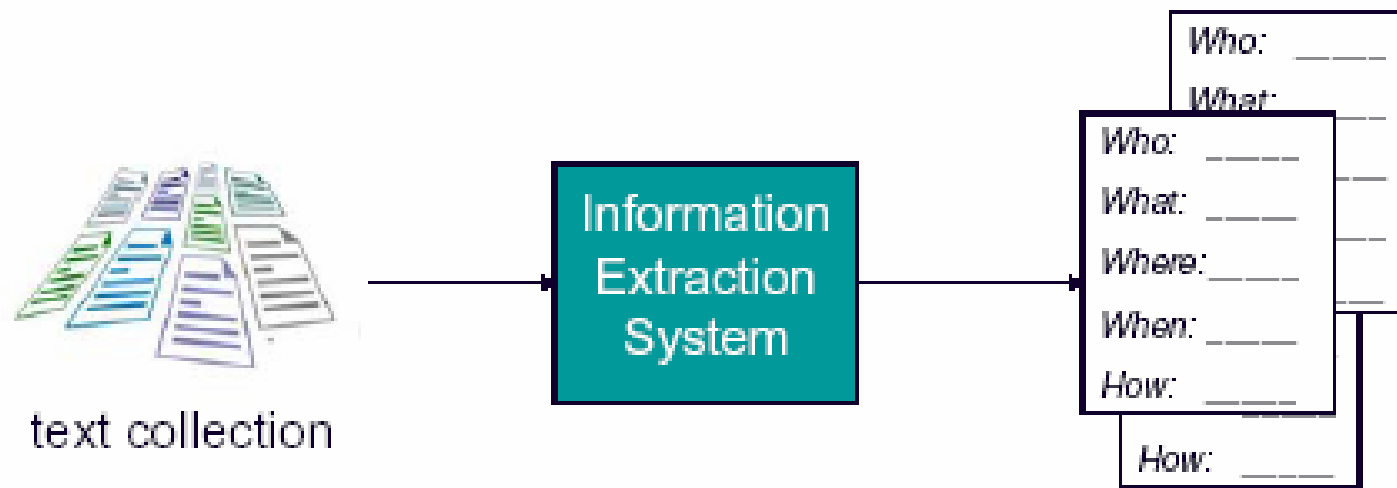
- How do we estimate weights?
  - Generalized Iterative Scaling (Darroch&Ratcliff,'72)
  - Improved Iterative Scaling (Della Pietra et al, '97)

# Advantage of MaxEnt

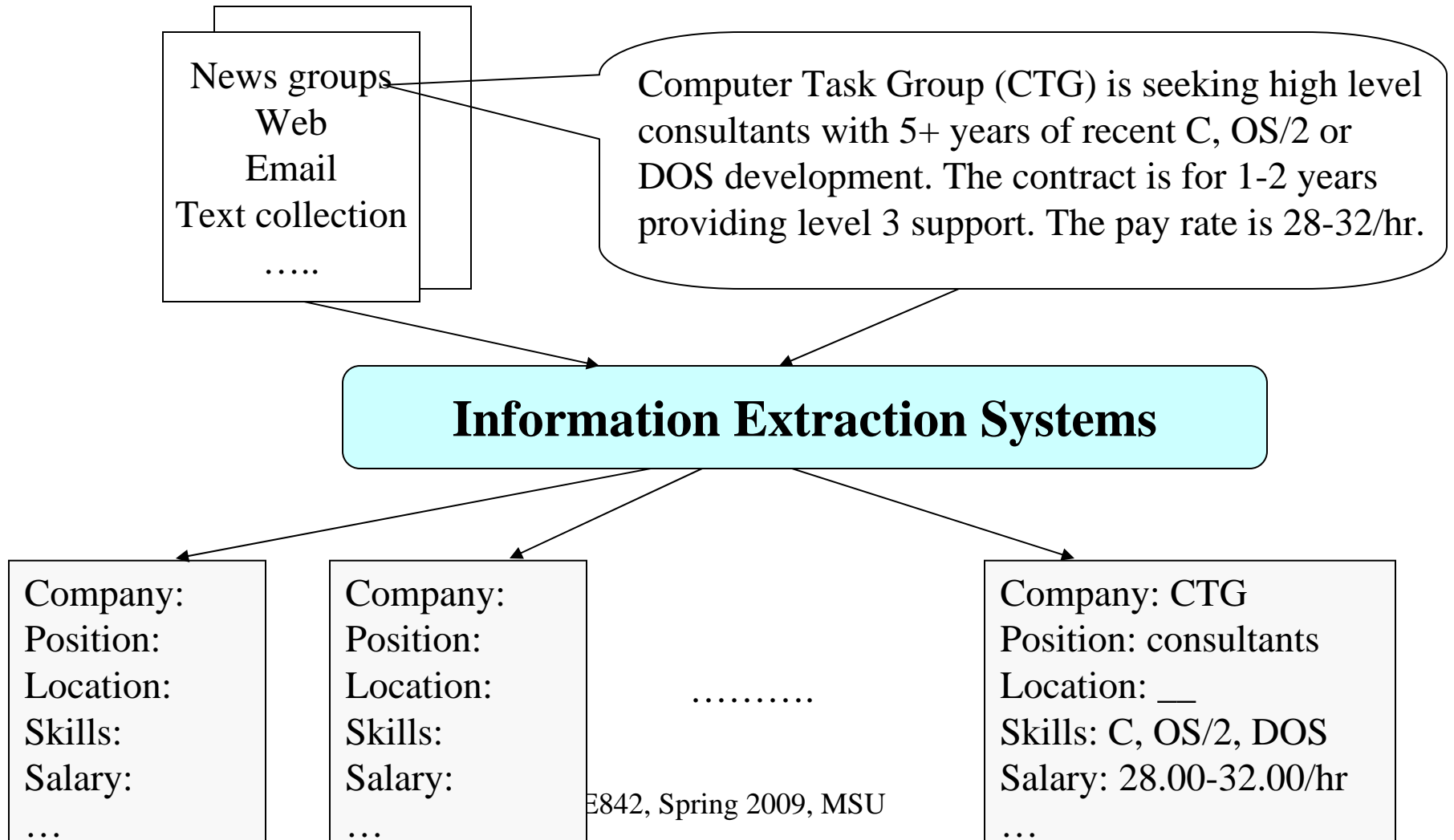
- Diverse forms of evidence
- No independence assumptions: contrast with Naïve Bayes
- Feature weights are determined automatically
- Empirically work well on many NLP problems

# Information Extraction

# What is Information Extraction



# What is Information Extraction



# Information Extraction (IE)

- Identify specific pieces of information (data) in an unstructured or semi-structured textual document.
- Transform unstructured information in a corpus of documents or web pages into a structured database
- Applied to different types of text:
  - Newspaper articles
  - Scientific articles
  - Web pages
  - Newsgroup messages
  - Classified ads
  - Medical literatures

# An Example

**Computer Task Group (CTG) is seeking high level consultants with 5+ years of recent C, OS/2 or DOS development. The contract is for 1-2 years providing level 3 support. The pay rate is 28-32/hr.**

# An Example

**Computer Task Group (CTG) is seeking high level consultants with 5+ years of recent C, OS/2 or DOS development. The contract is for 1-2 years providing level 3 support. The pay rate is 28-32/hr.**

## Lexical Processing and Parsing

# An Example

*Company Name*

Computer Task Group (CTG) is seeking high level consultants with 5+ years of recent C, OS/2 or DOS development. The contract is for 1-2 years providing level 3 support. The pay rate is 28-32/hr.

*Software Name*

*Salary/Rate*

**Name Entity Recognition**

# An Example

Computer Task Group (CTG) is seeking high level consultants with 5+ years of recent C, OS/2 or DOS development. The contract is for 1-2 years providing level 3 support. The pay rate is 28-32/hr.

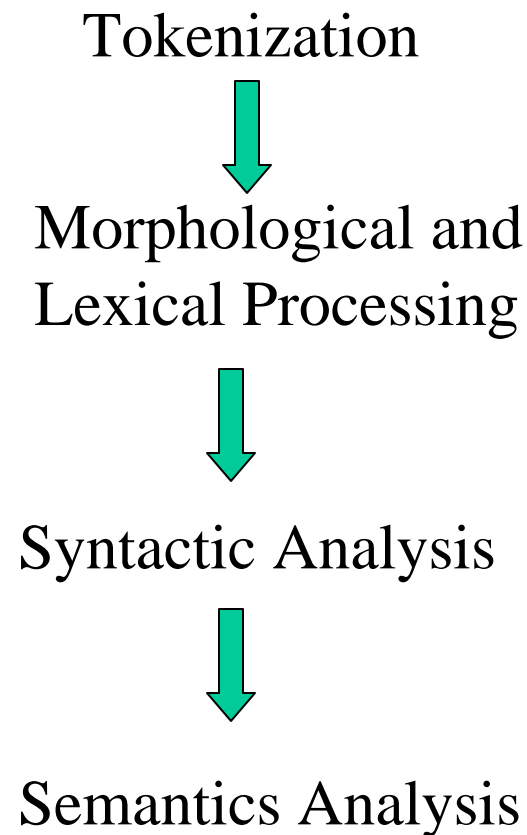
Company: CTG  
Position: consultants  
Location: \_\_\_  
Skills: C, OS/2, DOS  
Salary: 28.00-32.00/hr

...

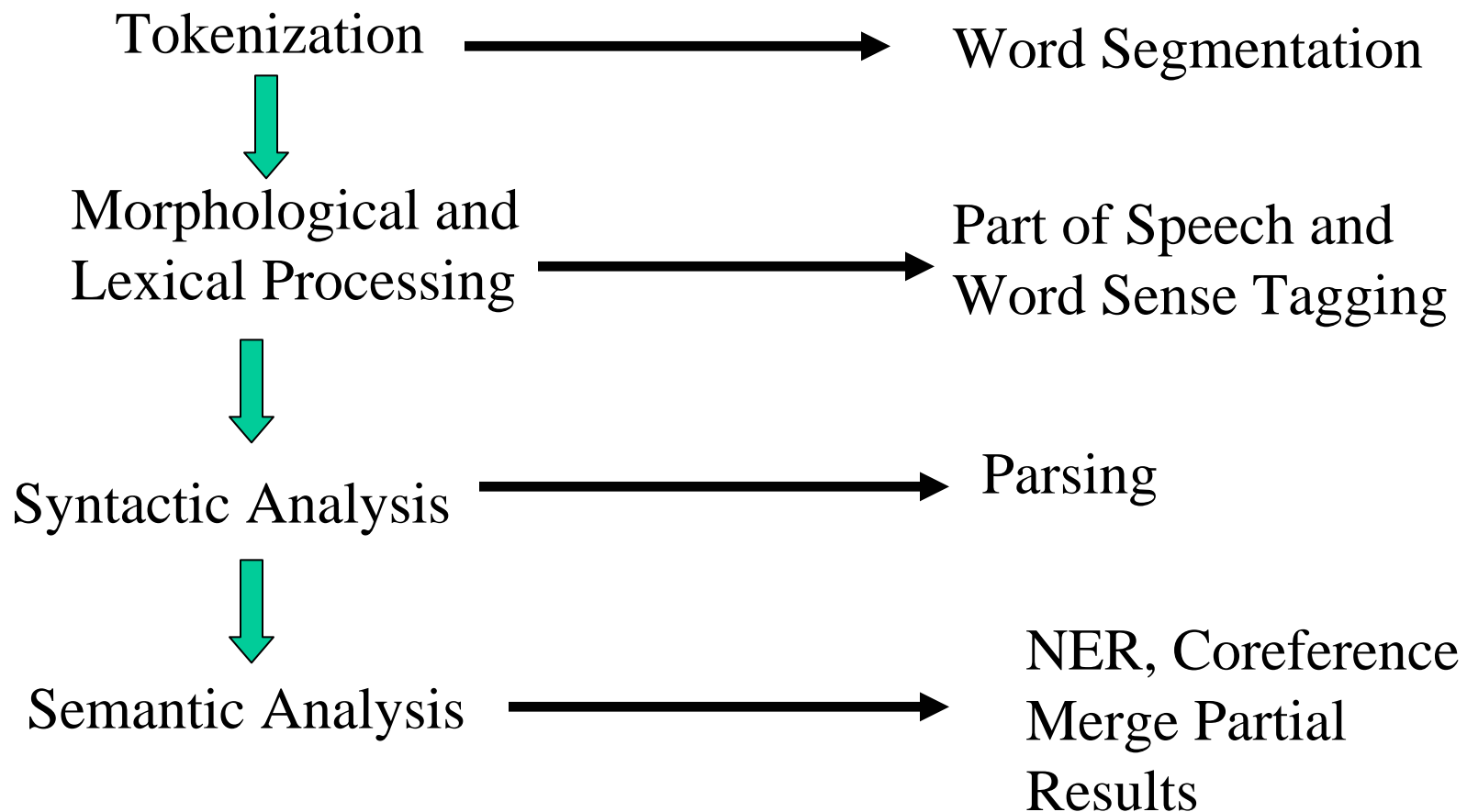
# Evaluation Metrics

- Precision and Recall:
  - Precision: correct answers / answers produced
  - Recall: correct answers / total possible correct answers
- F-measure: 
$$F = \frac{(\beta^2 + 1)P * R}{(\beta^2 P + R)}$$
  - Where  $\beta$  is a parameter representing relative importance of P and R.

# A Bare Bone Extraction System



# Flesh for the Bones



# Two Approaches

- Knowledge Engineering Approach
  - Grammars constructed by hand
  - Domain patterns discovered by introspection and corpus examination
  - Laborious tuning and hill-climbing
- Learning and Statistical Approach
  - Apply statistical methods where possible
  - Learn rules from annotated corpora
  - Learn from interaction with user

# Knowledge Engineering Approach

- Advantages
  - Skilled computational linguists can build good systems
- Disadvantages
  - Very laborious development process
  - Difficult to port systems to new domains
  - Requires expertise

# Learning-Statistical Approach

- Advantages
  - Domain portability is straightforward
  - Minimal expertise required for customization
  - Rule acquisition is data driven – complete coverage of examples
- Disadvantages
  - Training data may not exist and may be difficult or expensive to obtain

# A Combined Approach

- Use statistical methods on modules where training data exists, and high accuracy can be achieved
  - Part of Speech tagging
  - Name entity identification
  - Co-reference
- Use knowledge engineering when training data is sparse and human ingenuity is required
  - Domain processing

# Name Entity Recognition

- Named Entities are proper names in texts, i.e. the names of persons, organizations, locations, times and quantities.
- NER is the task of processing a text and identifying named entities

# Name Entity Recognition

Why is Named Entity Recognition difficult?

- Names too numerous to include in dictionaries
- Variations  
e.g. John Smith, Mr Smith, John
- Changing constantly  
new names invent unknown words
- Ambiguity  
For some proper nouns it is hard to determine the category

# Example

Delimit the named entities in a text and tag them with  
NE

Categories:

- entity names - ENAMEX
- temporal expressions - TIMEX
- number expressions - NUMEX

Subcategories of tags

- captured by a SGML tag attribute called TYPE

# Example

- Original text:

The U.K. satellite television broadcaster said its subscriber base grew 17.5 percent during the past year to 5.35 million

- Tagged text:

The `<ENAMEX TYPE="LOCATION">U.K.</ENAMEX>` satellite television broadcaster said its subscriber base grew `<NUMEX TYPE="PERCENT">17.5 percent</NUMEX>` during `<TIMEX TYPE="DATE">the past year</TIMEX>` to 5.35 million

# Maximum Entropy for NER

(Borthwick et al., 1998)

- Outcomes:  $N$  classes for MUC7 ( $N = 7$ )
  - Person Name, Company Name, Date
- For a particular class  $x$ :  $x\_start$ ,  $x\_continue$ ,  $x\_end$ ,  $x\_unique$
- $4N+1$  tags (29 tags)

[Jerry	Lee	Lewis	flew	to	Paris]
Per_start	Per_cont.	Per_end	other	other	loc_unique

# Types of Features

- Binary features
  - Token properties which are either on or off for a given token (e.g., All-caps, 2-digit-number, only-digits, initial-cap)
  - Overlapping allowed

$$f(a,b) = \begin{cases} 1: & \text{if } \textit{current\_token\_capitalized}(b) = \textit{true} \\ & \text{and } a = \textit{location\_start} \\ 0: & \textit{else} \end{cases}$$

# Types of Features

- Lexical features
  - Lexical lookup for words in the context for a current token
  - Lexicon is built automatically (just build a vocabulary  $V$  as “all words  $w: c(w) > 2$ ”)

$$f(a,b) = \begin{cases} 1: & \text{if } \textit{Lexical\_View}(\textit{token}_{-1}(b)) = \textit{Mr} \\ & \text{and } a = \textit{person\_unique} \\ 0: & \textit{else} \end{cases}$$

# Types of Features

- Section features
  - The current section of the article: title, textbody
- Dictionary features
  - Multi-words entries of pre-classified NE words (e.g., Michigan State University)
- External Systems Feature: use other taggers

$$f(a,b) = \begin{cases} 1 : \text{if } SystemA(token_0(b)) = person\_unique \\ \quad \text{and } a = person\_unique \\ 0 : \text{else} \end{cases}$$

# Feature Selection

- Put all possible features from the classes to be included into the model into a feature pool
  - Lexical features for range  $w_{-2}, \dots, w_0, \dots, w_2$ , vocabulary size of  $V$ , then  $5 \cdot (V+1) \cdot 29$  lexical features.
- Select all features which fire at least three times on the training corpus
- Features which predict the tag *other* have to fire six times to be included
- Lexical features which activate on  $w_{-2}$  and  $w_2$  are excluded if they predict *other*

# Evaluation

- F-measurement: 97.12% (used three other systems as external features)
- Human performance: 96.95-97.60.

# Current Performance of Supervised Methods

- CONLL 2003 shared tasks on name entity identification
  - Maximum Entropy Method (5)
  - Hidden Markov Model (4)
  - Connectionist Method (4)
- <http://cnts.uia.ac.be/conll2003/ner>

# Current Performance of Supervised Methods

English	precision	recall	F
[FIJZ03]	88.99%	88.54%	88.76±0.7
[CN03]	88.12%	88.51%	88.31±0.7
[KSNM03]	85.93%	86.21%	86.07±0.8
[ZJ03]	86.13%	84.88%	85.50±0.9
[CMP03b]	84.05%	85.96%	85.00±0.8
[CC03]	84.29%	85.50%	84.89±0.9
[MMP03]	84.45%	84.90%	84.67±1.0
[CMP03a]	85.81%	82.84%	84.30±0.9
[ML03]	84.52%	83.55%	84.04±0.9
[BON03]	84.68%	83.18%	83.92±1.0
[MLP03]	80.87%	84.21%	82.50±1.0
[WNC03]*	82.02%	81.39%	81.70±0.9
[WP03]	81.60%	78.05%	79.78±1.0
[HV03]	76.33%	80.17%	78.20±1.0
[DD03]	75.84%	78.13%	76.97±1.2
[Ham03]	69.09%	53.26%	60.15±1.3
baseline	71.91%	50.90%	59.61±1.2

German	precision	recall	F
[FIJZ03]	83.87%	63.71%	72.41±1.3
[KSNM03]	80.38%	65.04%	71.90±1.2
[ZJ03]	82.00%	63.03%	71.27±1.5
[MMP03]	75.97%	64.82%	69.96±1.4
[CMP03b]	75.47%	63.82%	69.15±1.3
[BON03]	74.82%	63.82%	68.88±1.3
[CC03]	75.61%	62.46%	68.41±1.4
[ML03]	75.97%	61.72%	68.11±1.4
[MLP03]	69.37%	66.21%	67.75±1.4
[CMP03a]	77.83%	58.02%	66.48±1.5
[WNC03]	75.20%	59.35%	66.34±1.3
[CN03]	76.83%	57.34%	65.67±1.4
[HV03]	71.15%	56.55%	63.02±1.4
[DD03]	63.93%	51.86%	57.27±1.6
[WP03]	71.05%	44.11%	54.43±1.4
[Ham03]	63.49%	38.25%	47.74±1.5
baseline	31.86%	28.89%	30.30±1.3