Information Extraction: Sequence Models, Information Extraction Tasks and Information Integration

CS 224N

2010

Feature-Based Classifiers

- Exponential (log-linear, maxent, logistic, Gibbs) models:
  - Have features $f_i: C \times D \rightarrow \mathbf{R}$, with weights λ_i, often indicator functions of a condition and class $f_i(c, d) \equiv [\Phi_k(d) \wedge c = c_j]$
  - Use the linear combination $\Sigma\lambda_i f_i(c,d)$ to produce a probabilistic model:

$$P(c \mid d, \lambda) = \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c'} \exp \sum_i \lambda_i f_i(c',d)}$$

Makes votes positive.

Normalizes votes.

  - The weights are the parameters of the probability model, combined via a "soft max" function
  - We choose parameters $\{\lambda_i\}$ that *maximize the conditional likelihood* of the data according to this model.

Feature Overlap

- Maxent models handle overlapping features well.
- Unlike a NB model, there is no double counting!

Empirical

	A	a
B	2	1
b	2	1

	A	a
B		
b		

All = 1

	A	a
B		
b		

A = 2/3

	A	a
B		
b		

A = 2/3

	A	a
B	1/4	1/4
b	1/4	1/4

	A	a
B	1/3	1/6
b	1/3	1/6

	A	a
B	1/3	1/6
b	1/3	1/6

	A	a
B		
b		

	A	a
B	λ_A	
b	λ_A	

	A	a
B	$\lambda'_A+\lambda''_A$	
b	$\lambda'_A+\lambda''_A$	

Example: NER Overlap

Grace is correlated with PERSON, but does not add much evidence on top of already knowing prefix features.

Feature Weights

Feature Type	Feature	PERS	LOC
Previous word	*at*	-0.73	0.94
Current word	*Grace*	0.03	0.00
Beginning bigram	*<G*	0.45	-0.04
Current POS tag	NNP	0.47	0.45
Prev and cur tags	IN NNP	-0.10	0.14
Previous state	Other	-0.70	-0.92
Current signature	Xx	0.80	0.46
Prev state, cur sig	O-Xx	0.68	0.37
Prev-cur-next sig	x-Xx-Xx	-0.69	0.37
P. state - p-cur sig	O-x-Xx	-0.20	0.82
…			
Total:		**-0.58**	**2.68**

Local Context

	Prev	Cur	Next
State	Other	???	???
Word	at	Grace	Road
Tag	IN	NNP	NNP
Sig	x	Xx	Xx

Feature Interaction

- Maxent models handle overlapping features well, but do not automatically model feature interactions.

Empirical

	A	a
B	1	1
b	1	0

	A	a
B		
b		

All = 1

	A	a
B		
b		

A = 2/3

	A	a
B		
b		

B = 2/3

	A	a
B	1/4	1/4
b	1/4	1/4

	A	a
B	1/3	1/6
b	1/3	1/6

	A	a
B	4/9	2/9
b	2/9	1/9

	A	a
B	0	0
b	0	0

	A	a
B	λ_A	
b	λ_A	

	A	a
B	$\lambda_A+\lambda_B$	λ_B
b	λ_A	

Feature Interaction

- If you want interaction terms, you have to add them:

Empirical

	A	a
B	1	1
b	1	0

	A	a
B		
b		

A = 2/3

	A	a
B		
b		

B = 2/3

	A	a
B		
b		

AB = 1/3

	A	a
B	1/3	1/6
b	1/3	1/6

	A	a
B	4/9	2/9
b	2/9	1/9

	A	a
B	1/3	1/3
b	1/3	0

- A disjunctive feature would also have done it (alone):

	A	a
B		
b		

	A	a
B	1/3	1/3
b	1/3	0

Feature Interaction

- For loglinear/logistic regression models in statistics, it is standard to do a greedy stepwise search over the space of all possible interaction terms.
- This combinatorial space is exponential in size, but that's okay as most statistics models only have 4–8 features.
- In NLP, our models commonly use hundreds of thousands of features, so that's not okay.
- Commonly, interaction terms are added by hand based on linguistic intuitions.

Example: NER Interaction

Previous-state and current-signature have interactions, e.g. P=PERS-C=Xx indicates C=PERS much more strongly than C=Xx and P=PERS independently.

This feature type allows the model to capture this interaction.

Local Context

	Prev	Cur	Next
State	Other	???	???
Word	at	Grace	Road
Tag	IN	NNP	NNP
Sig	x	Xx	Xx

Feature Weights

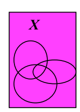
Feature Type	Feature	PERS	LOC
Previous word	*at*	-0.73	0.94
Current word	*Grace*	0.03	0.00
Beginning bigram	*<G*	0.45	-0.04
Current POS tag	NNP	0.47	0.45
Prev and cur tags	IN NNP	-0.10	0.14
Previous state	Other	-0.70	-0.92
Current signature	Xx	0.80	0.46
Prev state, cur sig	O-Xx	0.68	0.37
Prev-cur-next sig	x-Xx-Xx	-0.69	0.37
P. state - p-cur sig	O-x-Xx	-0.20	0.82
…			
Total:		**-0.58**	**2.68**

Classification

- What do these joint models of $\mathrm{P}(X)$ have to do with conditional models $\mathrm{P}(C|D)$?
- Think of the space $C \times D$ as a complex X.
  - C is generally small (e.g., 2-100 topic classes)
  - D is generally huge (e.g., space of documents)
- We can, in principle, build models over $\mathrm{P}(C,D)$.
- This will involve calculating expectations of features (over $C \times D$):

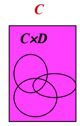
$$E(f_i) = \sum_{(c,d)\in(C,D)} P(c,d) f_i(c,d)$$

- Generally impractical: can't enumerate d efficiently.



C

D



Classification II

- D may be huge or infinite, but only a few d occur in our data.
- What if we add one feature for each d and constrain its expectation to match our empirical data?

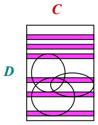
$$\forall (d) \in D \quad P(d) = \hat{P}(d)$$

- Now, most entries of $\mathrm{P}(c,d)$ will be zero.
- We can therefore use the much easier sum:

$$E(f_i) = \sum_{(c,d)\in(C,D)} P(c,d) f_i(c,d)$$
$$= \sum_{(c,d)\in(C,D) \wedge \hat{P}(d)>0} P(c,d) f_i(c,d)$$

C

D



Classification III

- But if we've constrained the D marginals

$$\forall (d) \in D \quad P(d) = \hat{P}(d)$$

then the only thing that can vary is the conditional distributions:

$$P(c,d) = P(c\,|\,d)P(d)$$
$$= P(c\,|\,d)\hat{P}(d)$$

- This is the connection between joint and conditional maxent / exponential models:
  - Conditional models can be thought of as joint models with marginal constraints.
- Maximizing joint likelihood and conditional likelihood of the data in this model are equivalent!

Easy Quiz Question!

- Suppose we have a 1 feature maxent model built over observed data as shown.
- What is the constructed model's probability distribution over the four possible outcomes?

Empirical

	A	a
B	2	1
b	2	1

Features

	A	a
B		
b		

Expectations

Probabilities

	A	a
B	(i)	(ii)
b	(iii)	(iv)

Issues of Scale

- Lots of features:
  - NLP maxent models can have millions of features.
  - Even storing a single array of parameter values can have a substantial memory cost.
- Lots of sparsity:
  - Overfitting very easy – need smoothing!
  - Many features seen in training will never occur again at test time.
- Optimization problems:
  - Feature weights can be infinite, and iterative solvers can take a long time to get to those infinities.

Smoothing: Issues

- Assume the following empirical distribution:

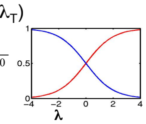
Heads	Tails
h	t

- Features: {Heads}, {Tails}
- We'll have the following model distribution:

$$p_{\text{HEADS}} = \frac{e^{\lambda_H}}{e^{\lambda_H} + e^{\lambda_T}} \qquad p_{\text{TAILS}} = \frac{e^{\lambda_T}}{e^{\lambda_H} + e^{\lambda_T}}$$

- Really, only one degree of freedom ($\lambda = \lambda_H - \lambda_T$)

$$p_{\text{HEADS}} = \frac{e^{\lambda_H} e^{-\lambda_T}}{e^{\lambda_H} e^{-\lambda_T} + e^{\lambda_T} e^{-\lambda_T}} = \frac{e^{\lambda}}{e^{\lambda} + e^{0}} \qquad p_{\text{TAILS}} = \frac{e^{0}}{e^{\lambda} + e^{0}}$$

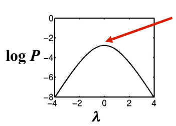


Smoothing: Issues

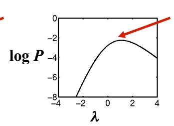
- The data likelihood in this model is:

$$\log P(h, t \mid \lambda) = h \log p_{\text{HEADS}} + t \log p_{\text{TAILS}}$$

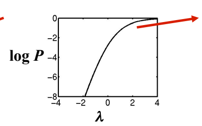
$$\log P(h, t \mid \lambda) = h\lambda - (t + h) \log (1 + e^{\lambda})$$



Heads	Tails
2	2



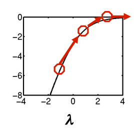
Heads	Tails
3	1



Heads	Tails
4	0

Smoothing: Early Stopping

- In the 4/0 case, there were two problems:
  - The optimal value of λ was ∞, which is a long trip for an optimization procedure.
  - The learned distribution is just as spiked as the empirical one – no smoothing.
- One way to solve both issues is to just stop the optimization early, after a few iterations.
  - The value of λ will be finite (but presumably big).
  - The optimization won't take forever (clearly).
  - Commonly used in early maxent work.



Heads	Tails
4	0

Input

Heads	Tails
1	0

Output

Smoothing: Priors (MAP)

- What if we had a prior expectation that parameter values wouldn't be very large?
- We could then balance evidence suggesting large parameters (or infinite) against our prior.
- The evidence would never totally defeat the prior, and parameters would be smoothed (and kept finite!).
- We can do this explicitly by changing the optimization objective to maximum posterior likelihood:

$$\log P(C, \lambda \mid D) = \log P(\lambda) + \log P(C \mid D, \lambda)$$

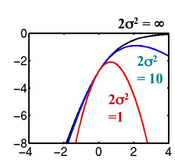
Posterior **Prior** **Evidence**

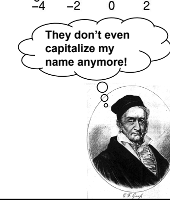
Smoothing: Priors

- Gaussian, or quadratic, or L2 priors:
  - Intuition: parameters shouldn't be large.
  - Formalization: prior expectation that each parameter will be distributed according to a gaussian with mean μ and variance σ^2.

$$P(\lambda_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(-\frac{(\lambda_i - \mu_i)^2}{2\sigma_i^2} \right)$$

  - Penalizes parameters for drifting to far from their mean prior value (usually μ=0).
  - $2\sigma^2$=1 works okay.





Example: NER Smoothing

Because of smoothing, the more common prefix and single-tag features have larger weights even though entire-word and tag-pair features are more specific.

Local Context

	Prev	Cur	Next
State	Other	???	???
Word	at	Grace	Road
Tag	IN	NNP	NNP
Sig	x	Xx	Xx

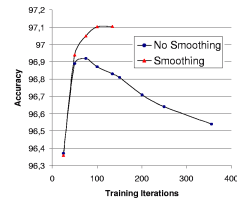
Feature Weights

Feature Type	Feature	PERS	LOC
Previous word	at	-0.73	0.94
Current word	Grace	0.03	0.00
Beginning bigram	<G	0.45	-0.04
Current POS tag	NNP	0.47	0.45
Prev and cur tags	IN NNP	-0.10	0.14
Previous state	Other	-0.70	-0.92
Current signature	Xx	0.80	0.46
Prev state, cur sig	O-Xx	0.68	0.37
Prev-cur-next sig	x-Xx-Xx	-0.69	0.37
P. state - p-cur sig	O-x-Xx	-0.20	0.82
…			
Total:		-0.58	2.68

Example: POS Tagging

- From (Toutanova et al., 2003):

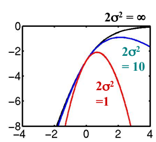
	Overall Accuracy	Unknown Word Acc
Without Smoothing	96.54	85.20
With Smoothing	97.10	88.20



- Smoothing helps:
  - Softens distributions.
  - Pushes weight onto more explanatory features.
  - Allows many features to be dumped safely into the mix.
  - Speeds up convergence (if both are allowed to converge)!

Smoothing: Priors

- If we use gaussian priors:
  - Trade off some expectation-matching for smaller parameters.
  - When multiple features can be recruited to explain a data point, the more common ones generally receive more weight.
  - Accuracy generally goes up!
- Change the objective (assume μ_i=0):



$$\log P(C,\lambda \mid D) = \log P(C \mid D,\lambda) \; - \log P(\lambda)$$

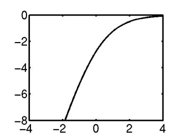
$$\log P(C,\lambda \mid D) = \sum_{(c,d)\in(C,D)} P(c \mid d,\lambda) \; - \sum_i \frac{\lambda_i^2}{2\sigma_i^2} + k$$

- Change the derivative:

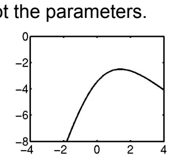
$$\partial \log P(C,\lambda \mid D) / \partial \lambda_i = \text{actual}(f_i, C) - \text{predicted}(f_i,\lambda) \quad -\lambda_i/\sigma^2$$

Smoothing: Virtual Data

- Another option: smooth the data, not the parameters.
- Example:



Heads	Tails
4	0



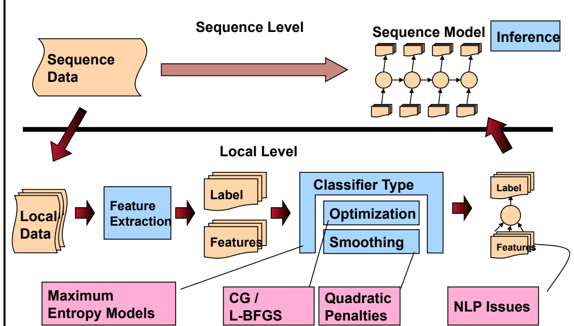
Heads	Tails
5	1

  - Equivalent to adding two extra data points.
  - Similar to add-one smoothing for generative models.
- Hard to know what artificial data to create!

Smoothing: Count Cutoffs

- In NLP, features with low empirical counts were usually dropped.
  - Very weak and indirect smoothing method.
  - Equivalent to locking their weight to be zero.
  - Equivalent to assigning them gaussian priors with mean zero and variance zero.
  - Dropping low counts does remove the features which were most in need of smoothing…
  - … and speeds up the estimation by reducing model size …
  - … but count cutoffs generally hurt accuracy in the presence of proper smoothing.
- We recommend: don't use count cutoffs unless absolutely necessary.

Sequence Inference



MEMM inference in systems

- For a Conditional Markov Model (CMM) a.k.a. a Maximum Entropy Markov Model (MEMM), the classifier makes a single decision at a time, conditioned on evidence from observations and previous decisions.
- A larger space of sequences is explored via search

Local Context

Decision Point

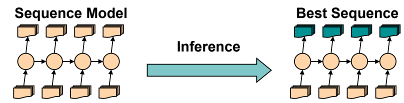
-3	-2	-1	0	+1
DT	NNP	VBD	???	???
The	Dow	fell	22.6	%

Features

W_0	22.6
W_{+1}	%
W_{-1}	fell
T_{-1}	VBD
T_{-1}-T_{-2}	NNP-VBD
hasDigit?	true
…	…

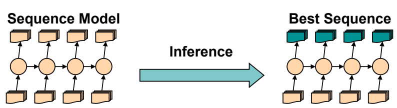
(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)

Two ways to Search: Beam Inference



- Beam inference:
  - At each position keep the top k complete sequences.
  - Extend each sequence in each local way.
  - The extensions compete for the k slots at the next position.
- Advantages:
  - Fast; and beam sizes of 3–5 are as good or almost as good as exact inference in many cases.
  - Easy to implement (no dynamic programming required).
- Disadvantage:
  - Inexact: the globally best sequence can fall off the beam.

Two ways to Search: Viterbi Inference



- Viterbi inference:
  - Dynamic programming or memoization.
  - Requires small window of state influence (e.g., past two states are relevant).
- Advantage:
  - Exact: the global best sequence is returned.
- Disadvantage:
  - Harder to implement long-distance state-state interactions (but beam inference tends not to successfully capture long-distance resurrection of sequences anyway).

Viterbi Inference: J&M Ch. 6

- I'm basically punting on this … read Ch. 6.
  - I'll do dynamic programming for parsing
- It's a small change from HMM Viterbi
  - From:

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, P(s_j|s_i)\, P(o_t|s_j) \quad 1 \le j \le N, 1 < t \le T$$

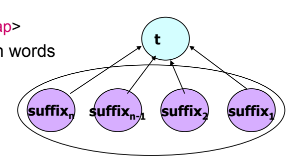
  - To:

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, P(s_j|s_i, o_t) \quad 1 \le j \le N, 1 < t \le T$$

Part-of-speech tagging: Generative HMM Tagging Models of Brants 2000

- Highly competitive with other state-of-the art models
- Trigram HMM with smoothed transition probabilities
- Capitalization feature becomes part of the state – each tag state is split into two e.g.
  NN → <NN,cap>,<NN,not cap>
- Suffix features for unknown words

$$P(w \mid tag) = P(suffix \mid tag)(w \mid suffix)$$
$$\approx \hat{P}(suffix)\tilde{P}(tag \mid suffix) / \hat{P}(tag)$$



$$\tilde{P}(tag \mid suffix_n) = \lambda_1 \hat{P}(tag \mid suffix_n) + \lambda_2 \hat{P}(tag \mid suffix_{n-1}) + \ldots + \lambda_n \hat{P}(tag)$$

MEMM Tagging Models -II

- Ratnaparkhi (1996): local distributions are estimated using maximum entropy models
  - Previous two tags, current word, previous two words, next two words, suffix, prefix, hyphenation, and capitalization features for unknown words
- Toutanova et al. (2003)
  - Richer features, bidirectional inference, better smoothing, better unknown word handling

Model	Overall Accuracy	Unknown Words
MEMM (Ratn. 1996)	96.63	85.56
HMM (Brants 2000)	96.7	85.5
MEMM (T. et al 2003)	97.24	89.04

CRFs [Lafferty, Pereira, and McCallum 2001]

- Another sequence model: Conditional Random Fields (CRFs)
- A whole-sequence conditional model rather than a chaining of local models.

$$P(c \mid d, \lambda) = \frac{\exp \sum_i \lambda_i f_i(c,d)}{\sum_{c'} \exp \sum_i \lambda_i f_i(c',d)}$$

- The space of c's is now the space of sequences
  - But if the features f_i remain local, the conditional sequence likelihood can still be calculated exactly using dynamic programming
- Training is slower, but CRFs avoid causal-competition biases
- These (or a variant using a max margin criterion) are seen as the state-of-the-art these days, and fairly standardly used

NER Results: CoNLL (2003) Named Entity Recognition task

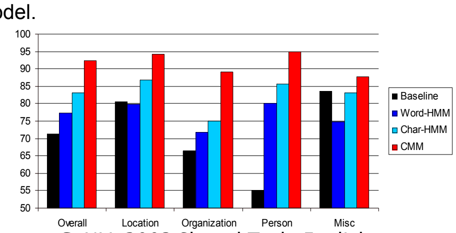
Task: Predict semantic label of each word in text

Foreign	NNP	I-NP	ORG
Ministry	NNP	I-NP	ORG
spokesman	NN	I-NP	O
Shen	NNP	I-NP	PER
Guofang	NNP	I-NP	PER
told	VBD	I-VP	O
Reuters	NNP	I-NP	ORG
:	:	:	:

Standard evaluation is per entity, *not* per token

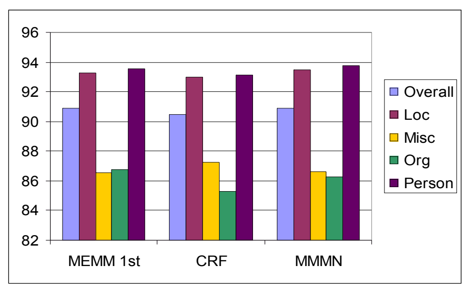
NER Results: Discriminative Model

- Increases from better features, a better classification model.

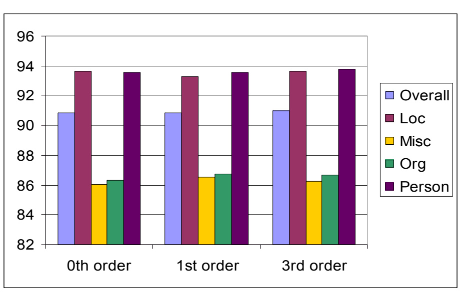


CoNLL 2003 Shared Task: English NER; entity precision/recall F1

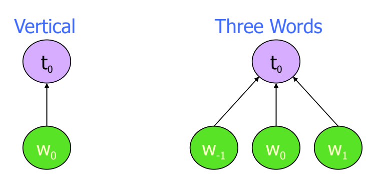
Sequence models? CoNLL 2003 NER shared task Results on English Devset



CoNLL NER Results: CMM Order



Sequence Tagging Without Sequence Information: POS tagging



Model	Features	Token	Unknown	Sentence
Vertical	56,805	**93.69%**	82.61%	26.74%
3Words	239,767	**96.57%**	86.78%	48.27%

Using 3 words works significantly better than using only the current word and the previous two or three tags instead! (Toutanova et al. 2003)

Biomedical NER Motivation

- The biomedical world has a huge body of information, which is growing rapidly.
  - MEDLINE, the primary research database serving the biomedical community, currently contains over 12 million abstracts, with 60,000 new abstracts appearing each month.
  - There is also an impressive number of biological databases containing information on genes, proteins, nucleotide and amino acid sequences, including *GenBank*, *Swiss-Prot*, and *Fly-Base*; each contains entries numbering from the thousands to the millions and are multiplying rapidly.
    - Currently, these resources are curated by hand by expert annotators at enormous expense.

Named Entity Recognition

- General NER vs. Biomedical NER

<PER> Christopher Manning </PER> is a professor at <ORG> Stanford University </ORG>, in <LOC> Palo Alto </LOC>.

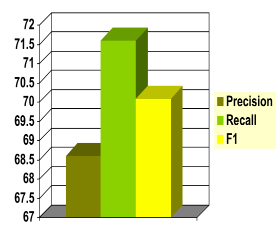
<RNA> TAR </RNA> independent transactivation by <PROTEIN> Tat </PROTEIN> in cells derived from the <CELL> CNS </CELL> - a novel mechanism of <DNA> HIV-1 gene </DNA> regulation.

Why is this difficult?

- The list of biomedical entities is growing.
  - New genes and proteins are constantly being discovered, so explicitly enumerating and searching against a list of known entities is not scalable.
  - Part of the difficulty lies in identifying previously unseen entities based on contextual, orthographic, and other clues.
- Biomedical entities don't have strict naming conventions.
  - Common English words such as *period*, *curved*, and *for* are used for gene names.
  - Entity names can be ambiguous. For example, in FlyBase, "clk" is the gene symbol for the "Clock" gene but it also is used as a synonym of the "period" gene.
- Biomedical entity names are ambiguous
  - Experts only agree on whether a word is even a gene or protein 69% of the time! (Krauthammer *et al.*, 2000)
  - Often systematic polysemies between gene, RNA, DNA, etc.

Finkel et al. (2004) Results

- BioNLP task − Identify genes, proteins, DNA, RNA, and cell types



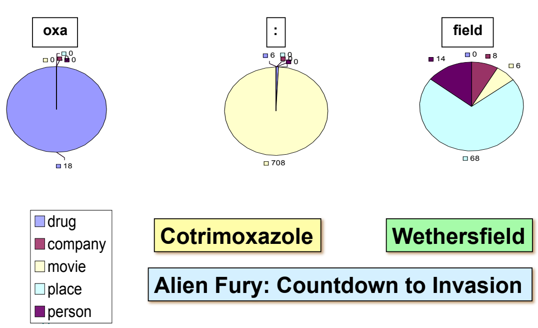
Precision	Recall	F1
68.6%	71.6%	70.1%

precision = tp / (tp + fp)

recall = tp / (tp + fn)

F1 = 2(precision)(recall) / (precision + recall)

Features: What's in a Name?



Cotrimoxazole

Wethersfield

Alien Fury: Countdown to Invasion

Interesting Features

- Word, and surrounding context
- Word Shapes
  - Map words to simplified representation that encodes attributes such as length, capitalization, numerals, Greek letters, internal punctuation, etc.

Varicella-zoster	Xx-xxx
mRNA	xXXX
CPA1	XXXd

- Character substrings

#Dpp# → #Dpp#, #Dpp, Dpp#, #Dp, Dpp, pp#, #D, Dp, pp, p#, D, p

The full task of "Information Extraction"

As a family of techniques:

Information Extraction = segmentation + classification + association + clustering

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

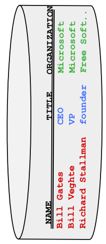
For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

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



Relation Extraction

May 19 1995, Atlanta -- The Centers for Disease Control and Prevention, which is in the front line of the world's response to the deadly Ebola epidemic in Zaire, is finding itself hard pressed to cope with the crisis…

Information Extraction System

Disease Outbreaks in *The New York Times*

Date	*Disease Name*	*Location*
Jan. 1995	Malaria	Ethiopia
July 1995	Mad Cow Disease	U.K.
Feb. 1995	Pneumonia	U.S.

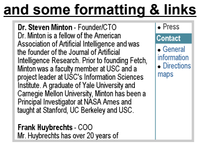
- Can be done by hand-written rules over text, perhaps annotated for NER, POS.
- Commonly done as classification decision based on identified entities and words/parse tree patterns between them
- Considerable work on bootstrapping learning from seed examples

Landscape of IE Tasks (1/4): Degree of Formatting

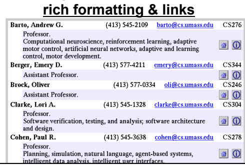
Text paragraphs without formatting

Astro Teller is the CEO and co-founder of BodyMedia. Astro holds a Ph.D. in Artificial Intelligence from Carnegie Mellon University, where he was inducted as a national Hertz fellow. His M.S. in symbolic and heuristic computation and B.S. in computer science are from Stanford University. His work in science, literature and business has appeared in international media from the New York Times to CNN to NPR.

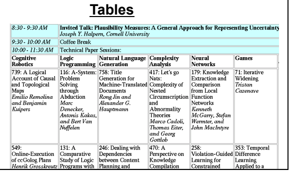
Grammatical sentences and some formatting & links



Non-grammatical snippets, rich formatting & links

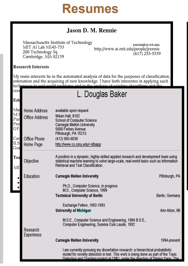


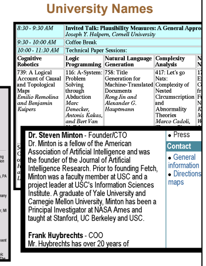
Tables



Landscape of IE Tasks (2/4): Intended Breadth of Coverage

Web site specific	Genre specific	Wide, non-specific
Formatting	Layout	Language
Amazon.com Book Pages	Resumes	University Names





Landscape of IE Tasks (3/4): Complexity

E.g. word patterns:

Closed set

U.S. states

He was born in Alabama…

The big Wyoming sky…

Regular set

U.S. phone numbers

Phone: (413) 545-1323

The CALD main office can be reached at 412-268-1299

Complex pattern

U.S. postal addresses

University of Arkansas
P.O. Box 140
Hope, AR 71802

Headquarters:
1128 Main Street, 4th Floor
Cincinnati, Ohio 45210

Ambiguous patterns, needing context and many sources of evidence

Person names

…was among the six houses sold by Hope Feldman that year.

Pawel Opalinski, Software Engineer at WhizBang Labs.

Landscape of IE Tasks (4/4): Single Field/Record

Jack Welch will retire as CEO of General Electric tomorrow. The top role at the Connecticut company will be filled by Jeffrey Immelt.

Single entity

Person: Jack Welch

Person: Jeffrey Immelt

Location: Connecticut

Binary relationship

Relation: Person-Title
Person: Jack Welch
Title: CEO

Relation: Company-Location
Company: General Electric
Location: Connecticut

N-ary record

Relation: Succession
Company: General Electric
Title: CEO
Out: Jack Welsh
In: Jeffrey Immelt

"Named entity" extraction