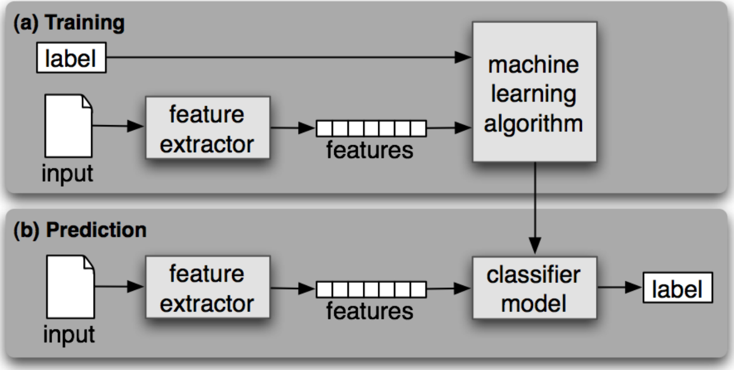
Information extraction: Named Entities

Vivi Nastase

with material from Marius Pasca's CIKM-2011 tutorial on IE

Summer semester 2012, ICL, University of Heidelberg

Machine learning – roughly



Evaluation (most frequently)

		Actual classification	
		positive	negative
Hypothesis	positive	true positive (tp)	false positive (fp)
	negative	false negative (fn)	true negative (tn)

Precision $P = \frac{TP}{TP+FP}$

Recall $R = \frac{TP}{TP+FN}$

Accuracy $A = \frac{TP+TN}{TP+FP+FN+TN}$

F-measure

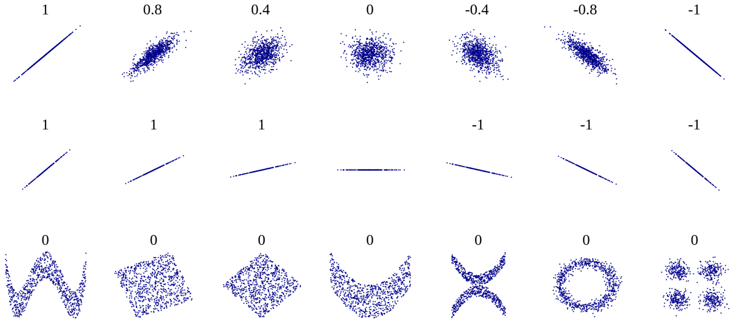
$$F = \frac{(1+\beta^2)PR}{\beta^2 P+R}$$

$$F = \frac{PR}{(1-\alpha)P+\alpha(R)}; \alpha = \frac{1}{1+\beta^2}$$

Most commonly used:

$$\beta = 1 \rightarrow F1 = \frac{2PR}{P+R}$$

Evaluation – correlation



Pearson's correlation coefficient

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})}}$$

Spearman's rank correlation

x_i, y_i – ranks of X_i, Y_i

$$\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

Named entities

Dr. Michael Jordan of the University of California Berkeley presents "Machine Learning from an Nonparametric Bayesian Point of View" March 27, 2008.

Named entity recognition

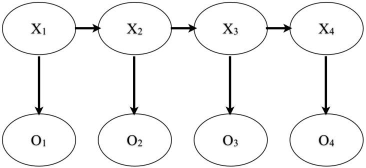
phrase classification

- predefine a set of entity types (Person, Organization, Location, ...)
- classify each phrase into one of the entity types + *non-entity*
- models = sets of extraction patterns

token classification

- classify = tagging each token as **I**nside or **O**utside a named entity
- extract contiguous tokens tagged **I** as named entities

NER with HMMs



Markov assumption: $P(x_i|x_{i-1}...x_1) = P(x_i|x_{i-1})$

Maximize $P(\mathbf{X}|\mathbf{O}, \lambda)$

$\mathbf{X} = x_1...x_T$ – sequence of hidden variable values

$\mathbf{O} = o_1...o_T$ – observations

$\mathbf{Q} = \{q_1, ..., q_N\}$ – possible states

$\lambda = (A, B)$

A (N x N)

$a_{ij} = p(x_i|x_j)$ transition probabilities

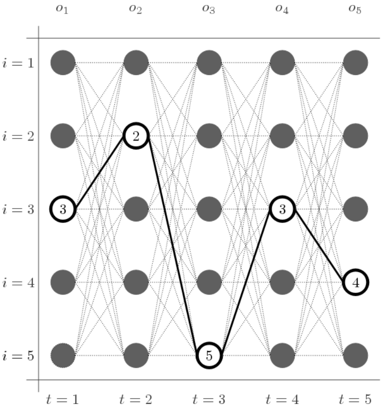
$a_{0j} = p(x_j)$ initial state probabilities

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$$

B (T) : $b_i(o_k) = p(o_k|x_i)$ emission probabilities

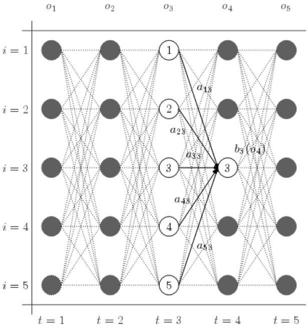
Sequence labeling with HMMs

The Viterbi Algorithm



$$v_1(j) = a_{0j} b_j(o_1) \quad j = 1, N$$
$$v_t(j) = max_i v_{t-1}(i) a_{ij} b_j(o_t) \quad j = 1, N$$
$$back(j) = argmax_i v_{t-1}(i) a_{ij} b_j(o_t)$$

Forward variable

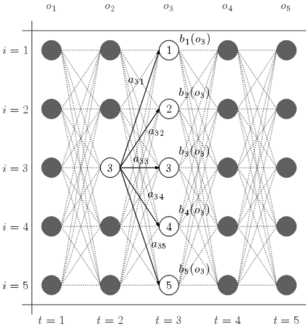


Forward variable = probability of being in state j after the first t observations

$\alpha_t(j) = P(o_1, ..., o_t, q_t = j|\lambda)$

$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)$

Backward variable



Backward variable = probability of seeing the observations $o_{t+1}, ..., o_T$ given that the state at time t is i

$\beta_t(i) = P(o_{t+1}, ..., o_T | q_t = i, \lambda)$

$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$

Learning an HMM – $\lambda = (A, B)$

Forward-backward algorithm

initialize A, B

iterate until convergence

E-step $\forall i, j, t$

$$\gamma_t(j) = P(q_t = j|\mathcal{O}, \lambda) = \frac{\alpha_t(j)\beta_t(j)}{P((O)|\lambda)}$$

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j|\mathcal{O}, \lambda) = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(N)}$$

M-step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{j=1}^{N} \xi_t(i,j)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t. o_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

return A, B

IOB sequence labeling with HMMs

- tokenize text
- split text into sentences (our sequences)
- hidden variable possible values: I, O, B
- estimate λ from an annotated corpus

$$a_{ij} = p(x_i | x_j) = \frac{c(x_i, x_j)}{c(x_j)}$$

$$b_j(o_t) = \frac{c(x_j, o_t)}{c(x_j)}$$

or learn λ using the forward-backward algorithm

Shortcomings of HMMs

- the output independence assumption – observations are independent from each other
- no general information (e.g. capitalization, POS)
- the inferred sequence of labels maximizes the likelihood $X = argmax_X P(O|X)$

Markov Random Fields

undirected graphical models $G = (V, E)$

- V = a set of vertices (corresp. to random variables)
- E = a set of undirected edges (corresp. to dependencies)
- $N(V_i)$ = the set of neighbours of vertex $V_i \in V$

Markov Random Field

$\forall V_i \in V, P(V_i | V - V_i) = P(V_i | N(V_i))$

Conditional Markov Random Field

$V = X \cup Y$

X = set of observed variables

Y = set of hidden variables

$\forall Y_i \in Y, P(Y_i | X, Y - Y_i) = P(Y_i | X, N(Y_i))$

Learning with Conditional Random Fields

$$P_\Lambda(\mathbf{x}|\mathbf{o}) = \frac{1}{Z_o} e^{\sum_{t=1}^{T} \sum_k \lambda_k f_k(x_{t-1}, x_t, \mathbf{o}, t)}$$

$Z_o = \sum_{x \in X^T} e^{\sum_{t=1}^{T} \sum_k \lambda_k f_k(x_{t-1}, x_t, \mathbf{o}, t)}$ – normalization factor

$f_k(x_{t-1}, x_t, \mathbf{o}, t)$ – feature function

$\Lambda = \{\lambda_1, ..., \lambda_K\}$
λ_k – learned weight for each feature function

Computing Z_o

Forward-backward algorithm with:

$$\alpha_{t+1}(x) = \sum_{x'} \alpha_t e^{\sum_k \lambda_k f_k(x', x, \mathbf{o}, t)}$$

$$Z_o = \sum_x \alpha_T(x)$$

Estimate feature weights

Training data

$\mathcal{D} = \{(o^{(l)}, y^{(l)})\}_{l=1}^{M}$
$o^{(l)} = (o_1^{(l)}, ..., o_T^{(l)})$
$y^{(l)} = (y_1^{(l)}, ..., y_T^{(l)})$

Conditional log-likelihood

$$\begin{aligned}
\hat{\Lambda} &= argmax_{\Lambda} P_{\Lambda}(y|x) \\
&= argmax_{\Lambda} logP_{\Lambda}(y|x) \\
&= argmax_{\Lambda} \sum_{l=1}^{M} logP_{\Lambda}(y^{(l)}|x^{(l)}) \\
&= ... \\
&= argmax_{\Lambda} \sum_{l=1}^{M} \sum_{l=1}^{M} logP_{\Lambda}(y^{(l)}|x^{(l)}) - \sum_{j=1}^{N} \frac{w_j^2}{2\sigma_j^2}
\end{aligned}$$

Features for NER with CRFs

begins-with-number	contains-question-mark
begins-with-punctuation	contains-question-word
begins-with-question-word	ends-with-question-mark
begins-with-subject	first-alpha-is-capitalized
blank	indented
contains-alphanum	only-punctuation
...	

Named entity disambiguation

Bunescu & Pasca, 2006 *Using encyclopedic knowledge for named entity disambiguation*

Ambiguity

Michael Jordan and his family moved from Brooklyn to Wilmington, N.C., when he was a small child and his famed basketball career has taken flight from ...

Michael Jeffrey Jordan (born February 17, 1963) is a retired American professional basketball player, active entrepreneur, and majority owner of the Charlotte Bobcats.

Dr. Michael Jordan of the University of California Berkeley presents "Machine Learning from an Nonparametric Bayesian Point of View" March 27, 2008.

Disambiguation approach

- Build a dictionary D of named entities
  - large scale – Wikipedia
  - map each name $d \in D$ to the set of entities $d.E$ in Wikipedia it can refer to
- Supervised disambiguation method:
  - detection – detect when a proper name refers to a named entity
  - disambiguation – find the correct referent given the context

Names in Wikipedia

- Assumption: Wikipedia articles describe concepts
- Names for Wikipedia articles:
  - article titles
  - redirect links
  - disambiguation articles
  - anchor texts of hyperlinks

Notations

e – entity (Wikipedia article)

$e.title$ – title name (e.g. *Michael Jordan (mycologist)*)

$e.name$ – clean name (e.g. *Michael Jordan*)

$e.T$ – text of the article (e.g. *United States*)

$e.R$ – set of all names that redirect to e (e.g. *USA, US, ...*)

$e.D$ – set of names whose disambiguation page contains a link to e (e.g. *US, America, ...*)

Named entity dictionary from Wikipedia

Collect all entity names that satisfy one of:

1. *e.title* is a multi-word term, and all content words are capitalized (e.g. *The Witches of Eastwick*)
2. *e.title* is a one-word term which contains at lest 2 capital letters (e.g. NATO)
3. at least 75% of the title occurrences inside the article are capitalized

Notation

$d \in D$ – a proper name entry in the dictionary D

$d.E$ – the set of entities whose name may be d

$e \in d.E \leftrightarrow e.name \vee d \in e.R \vee d \in e.D$

Disambiguation training data

The [[Vatican City|Vatican]] is now an enclave surrounded by [[Rome]].

Notation

$q.E$ – the set of entities associated with the query q in D

$q.e \in q.E$ – the true entity associated with q

$q.T$ – the text within a window of size n centered on q's hyperlink.

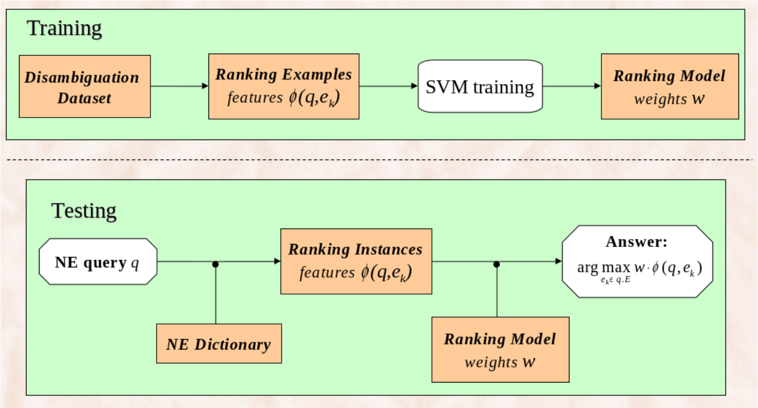
Disambiguation through ranking

- a scoring function that computes the compatibility between q and any of the potential referents e_k:
  $score(q, e_k) = w\phi(q, e_k) \quad \phi = [\phi_{cos}|\phi_{w,c}|\phi_{out}]$
  - context-article similarity:
    $\phi_{cos}(q, e_k) = cos(q.T, e_k.T) = \frac{q.T}{\|q.T\|}\frac{e_k.T}{\|e_k.T\|}$
    $\forall w \in q.T$ or $w \in e_k.T : d_w = f(w) ln \frac{N}{df(w)}$
  - category score, for each Wikipedia word w and category C pair:

$$\phi_{w,c}(q, e_k) = \begin{cases} 1 & \text{if } w \in q.T, c \in e_k.C \\ 0 & \text{otherwise} \end{cases}$$

  - special feature for *out-of-Wikipedia* entities:
    $\phi_{out}(q, e_k) = \delta(e_k, e_{out})$
- for an instance q, select $e = argmax_{e_k \in q.E} score(q, e_k)$

Disambiguation – system overview



Extracting information from open texts

Marius Pasca – CIKM tutorial on Information Extraction, 2011

Sources of open-domain information

Human compiled knowledge resources

- created by experts
- created collaboratively by non-experts

Sources of textual data

- text document (various degrees of structure)
- (Web) search queries

Expert-built resources

WordNet C. Fellbaum, 1998 *An Electronic Lexical Database*

- lexical database of English
- various extensions (languages, domains, sentiment)
- hypernym/hyponym and meronym/holonym hierarchies (and other relations)
- 155,000+ words / 117,000+ synsets

Cyc D. Lenat, 1995 *CYC: A large-scale investment in knowledge infrastructure*

- knowledge base of common-sense and encyclopedic knowledge
- concepts and relations organized in hierarchies
- 300,000+ concepts / 3+ million assertions

Collaborative non-expert resources

Open Mind P. Singh et al., 2002 *Open Mind Common Sense: Knowledge acquisition from the General Public*

- collect (common sense) knowledge in (simple) natural language
- 800,000+ facts in English

MindPixel knowledge base of millions of true/false or probabilistic propositions

Wikipedia M. Remy, 2002 *Wikipedia: The free encyclopedia*

- $\approx$ 4 million articles in English
- versions in 200+ languages

Collaborative non-expert resources

DBpedia C. Bizer et al., 2009 *DBpedia – A crystallization point for the web of data*

- converts information from Wikipedia's databases
- mappings from subset of Wikipedia infoboxes to ontology
- mappings from Wikipedia articles to WordNet
- 2.5+ million instances / 250+ million relations

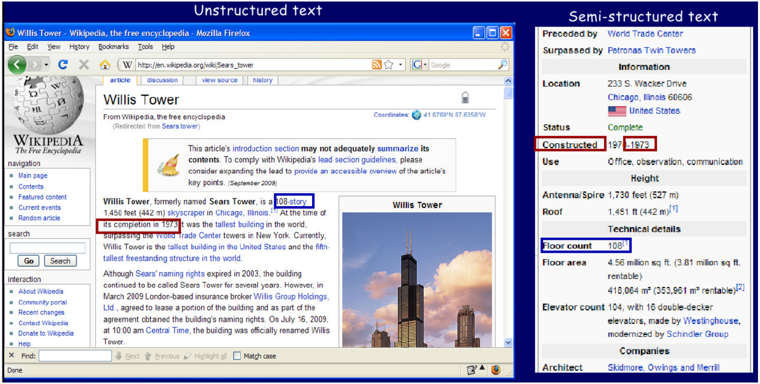
YAGO Suchanek et al., 2007 *YAGO – A core of semantic knowledge*

- semantic knowledge base derived from Wikipedia, WordNet, GeoNames
- 10+ million entities / 120+ million facts

Freebase K. Bollacker et al., 2008 *Freebase: A Collaboratively created graph database for structuring human knowledge*

- repository for storing structured data from Wikipedia, other sources, and additional user contributions
- 20+ million instances / 300+ million instances

Sources of textual data : documents



Sources of textual data : queries

Query logs:

- requests capture knowledge that the users already have
- the answers to requests capture knowledge the users don't yet have
- short length (2-3 words)
- low quality
- self-contained

Challenges in open-domain extraction

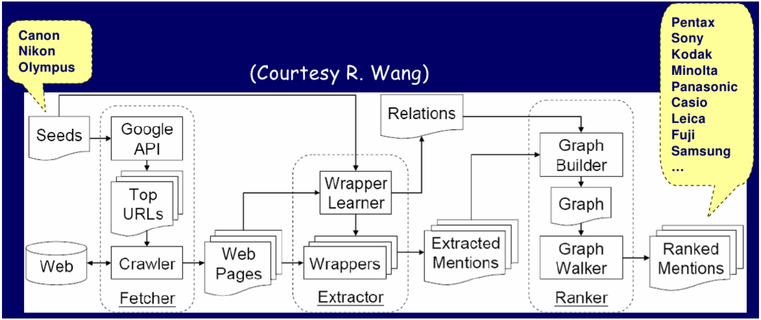
- **scale** – large text collections
  - time-efficient algorithms
  - shallow processing is preferred
- **diversity** – fine-grained classes of instances/relations
- **uncertainty** – use redundancy as a proxy for trustworthiness

Extraction methods

Methods for extraction of:

- concepts and instances
  - flat sets of unlabeled instances
  - flat sets of labeled instances
  - conceptual hierarchies
- relations and attributes
  - for flat concepts
  - for building ontologies

Extraction from web documents



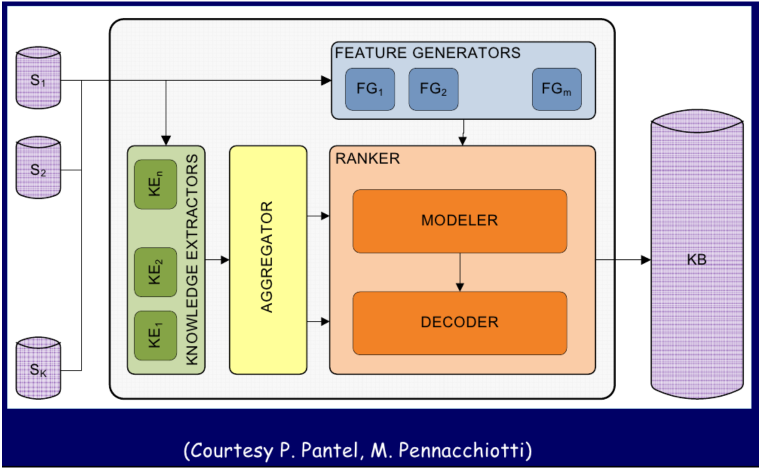
- start with seed instances
- submit queries, fetch Web documents with seed instances
- construct patterns for identifying more candidate instances
- rank candidate instances

Extraction from web documents

```
<li class="ford"><a href="http://www.curryauto.com/">
<img src="/common/logos/ford/logo-horiz-rgb-lg-dkbg.gif" alt="3"></a>
    <ul><li class="last"><a href="http://www.curryauto.com/">
        <span class="dName">Curry Ford</span>...</li></ul>
</li>
<li class="honda"><a href="http://www.curryauto.com/">
<img src="/common/logos/honda/logo-horiz-rgb-lg-dkbg.gif" alt="4"></a>
    <ul><li><a href="http://www.curryhonda-ga.com/">
        <span class="dName">Curry Honda Atlanta</span>...</li>
        <li><a href="http://www.curryhondamass.com/">
            <span class="dName">Curry Honda</span>...</li>
        <li class="last"><a href="http://www.curryhondany.com/">
            <span class="dName">Curry Honda Yorktown</span>...</li></ul>
</li>
<li class="acura"><a href="http://www.curryauto.com/">
<img src="/curryautogroup/images/logo-horiz-rgb-lg-dkbg.gif" alt="5"></a>
    <ul><li class="last"><a href="http://www.curryacura.com/">
        <span class="dName">Curry Acura</span>...</li></ul>
</li>
<li class="nissan"><a href="http://www.curryauto.com/">
<img src="/common/logos/nissan/logo-horiz-rgb-lg-dkbg.gif" alt="6"></a>
    <ul><li class="last"><a href="http://www.geisauto.com/">
        <span class="dName">Curry Nissan</span>...</li></ul>
</li>
```

(Courtesy R. Wang)

Extraction from multiple sources



Pantel & Pennacchiotti, 2006 *Entity extraction via Ensemble semantics*

- target classes, each with its own set of instances
- combine multiple data sources: web documents, queries, HTML tables, articles from Wikipedia

Features

Family	Type		Features
Web (w)	Frequency	(wF)	term frequency; document frequency; term frequency as noun phrase
	Pattern	(wP)	confidence score returned by KE_{pat}; pmi with the 100 most reliable patterns used by KE_{pat}
	Distributional	(wD)	distributional similarity with the centroid in KE_{dis}; distributional similarities with each seed in $\mathcal{S}$
	Termness	(wT)	ratio between term frequency as noun phrase and term frequency; pmi between internal tokens of the instance; capitalization ratio
Query log (q)	Frequency	(qF)	number of queries matching the instance; number of queries containing the instance
	Co-occurrence	(qC)	query log pmi with any seed in $\mathcal{S}$
	Pattern	(qP)	pmi with a set of trigger words $\mathcal{T}$ (i.e., the 10 words in the query logs with highest pmi with $\mathcal{S}$)
	Distributional	(qD)	distributional similarity with $\mathcal{S}$ (vector coordinates consist of the instance's pmi with the words in $\mathcal{T}$)
	Termness	(qT)	ratio between the two frequency features F
Web table (t)	Frequency	(tF)	table frequency
	Co-occurrence	(tC)	table pmi with $\mathcal{S}$; table pmi with any seed in $\mathcal{S}$
Wikipedia (k)	Frequency	(kF)	term frequency
	Co-occurrence	(kC)	pmi with any seed in $\mathcal{S}$
	Distributional	(kD)	distributional similarity with $\mathcal{S}$

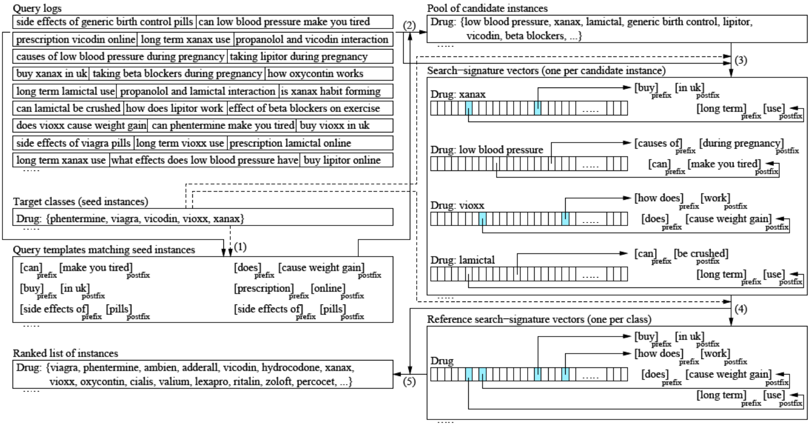
(Courtesy P. Pantel, M. Pennacchiotti)

Extracting NEs from query logs

Pasca, 2007 *Weakly-supervised discovery of named entities using Web search queries*

- **Input**
  - target classes, as sets of seeds: e.g. for Company – Honda, Oracle, Reuters, ...
- **Data** – anonymized search queries and their frequencies
- **Output** – ranked list of class instances

Extracting NEs from query logs



Vector representation

- signature-vector for candidate instance – aggregates all (weighted) prefixes and postfixes for the instance
- signature-vector for the class – adds the signature vectors for all instances
- rand each candidate instance based on its signature-vector similarity to the class signature vector:
  Jenses-Shannon divergence

$$JSD(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M)$$

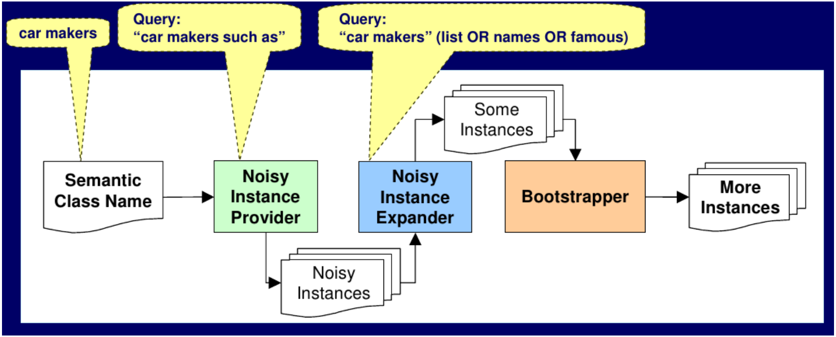
  where $M = \frac{1}{2}(P + Q)$
  Kullback-Leibler divergence

$$D_{\mathrm{KL}}(P \| Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}$$

Extracted patterns

Country:	*Drug*:	*VideoGame*:
what type of government does $\mathcal{I}$ have	how long does $\mathcal{I}$ stay in your system	how many copies of $\mathcal{I}$ have been sold
what do people in $\mathcal{I}$ eat	can $\mathcal{I}$ be crushed	how much does $\mathcal{I}$ cost
what is the weather like in $\mathcal{I}$ in march	what does the $\mathcal{I}$ pill look like	where can i play $\mathcal{I}$ online for free
how to apply for $\mathcal{I}$ visa	how much does $\mathcal{I}$ cost	how to install $\mathcal{I}$ mods
how did $\mathcal{I}$ gain independence	how does $\mathcal{I}$ affect the heart	when is $\mathcal{I}$ coming out on gamecube
where is $\mathcal{I}$ on the map	how is $\mathcal{I}$ manufactured	how many $\mathcal{I}$ levels
what continent is $\mathcal{I}$ on	does $\mathcal{I}$ cause weight gain	what copy protection does $\mathcal{I}$ use
what is $\mathcal{I}$'s currency	when was $\mathcal{I}$ fda approved	why does $\mathcal{I}$ crash
why did $\mathcal{I}$ join the eu	can $\mathcal{I}$ make you tired	how to add bots to $\mathcal{I}$ server
why is $\mathcal{I}$ poor	what $\mathcal{I}$ is made out of	who made $\mathcal{I}$ 2

Extracting instances within labeled concepts



Extracting instances within labeled concepts

Rule template

Predicate	Class1
Pattern	NP1 *such as* NP2
Constraints	head(NP1) = plural(label(Class1)) & properNoun(head(NP2))
Bindings	Class1(head(NP2))

↓

Extraction rule

Predicate	Car maker
Pattern	NP1 *such as* NP2
Constraints	head(NP1) = car makers & properNoun(head(NP2))
Bindings	Car maker(head(NP2))
Keywords	*car makers such as*

Extraction loop

Extractor
- convert rules to keyword-based queries
- submit queries to search engine
- apply rules to retrieved Web documents

Assessor
- estimate probability of correctness for each extraction
- compute association strength between extracted instances (e.g. *Kuala Lumpur* for *City*) and "discriminator" phrases (e.g. *city*)

Bootstrapping
- convert rule templates into extraction rules and discriminators
- select most productive extraction rules from previous iteration
- apply Extractor, then Assessor
- add candidate extractions to output
- repeat until all extraction rules and/or queries have been used