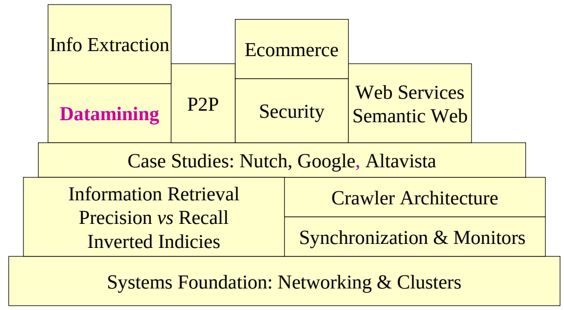
Issues in Datamining

CSE 454

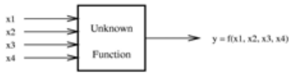
1

Course Overview



2

A Learning Problem



Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Learning occurs when PREJUDICE meets DATA!

- The nice word for prejudice is "**bias**".
- What kind of hypotheses will you consider?
  - What is allowable ***range*** of functions you use when approximating?
  - E.g. conjunctions
- What kind of hypotheses do you *prefer*?

4

Learning for Text Categorization

- Manual development of text categorization functions is difficult.
- Learning Algorithms:
  - **Bayesian (naïve)**
  - Neural network
  - **Relevance Feedback (Rocchio)**
  - Rule based (C4.5, Ripper, Slipper)
  - **Nearest Neighbor (case based)**
  - Support Vector Machines (SVM)

5

Bayesian Categorization

- Let set of categories be $\{c_1, c_2, \ldots c_n\}$
- Let E be description of an instance.
- Determine category of E by determining for each c_i

$$P(c_i \mid E) = \frac{P(c_i)P(E \mid c_i)}{P(E)}$$

- $P(E)$ can be determined since categories are complete and disjoint.

$$\sum_{i=1}^{n} P(c_i \mid E) = \sum_{i=1}^{n} \frac{P(c_i)P(E \mid c_i)}{P(E)} = 1$$

$$P(E) = \sum_{i=1}^{n} P(c_i)P(E \mid c_i)$$

6

Naïve Bayesian Categorization

- Too many possible instances (exponential in m) to estimate all $P(E \mid c_i)$
- If we assume features of an instance are independent given the category (c_i) (*conditionally independent*).

$$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$

- Therefore, we then only need to know $P(e_j \mid c_i)$ for each feature and category.

7

Evidence is Easy?

$$P(c_i \mid E) = \frac{\#}{\# \; + \#}$$

•Or…. Are their problems?

Smooth with a Prior

$$P(c_i \mid E) = \frac{\# \; + mp}{\# \; + \# \; + m}$$

p = prior probability
m = weight

Note that if m = 10, it means "I've seen 10 samples that make me believe $P(X_i \mid S) = p$"

Hence, m is referred to as the
equivalent sample size

Probabilities: Important Detail!

- $P(spam \mid E_1 \ldots E_n) = \prod_i P(spam \mid E_i)$

Any more potential problems here?

- We are multiplying lots of small numbers
  Danger of underflow!
  - $0.5^{57} = 7$ E -18
- Solution? Use logs and add!
  - $p_1 * p_2 = e^{\log(p1)+\log(p2)}$
  - Always keep in log form

Outline

- Evaluation of learning algorithms
- Co-training
- Focussed crawling

11

Evaluating Categorization

- Evaluation must be done on test data that are independent of the training data
  (usually a disjoint set of instances).
- *Classification accuracy*: c/n where
  - n is the ***total*** number of test instances,
  - c is the number of ***correctly classified*** test instances.
- Results can vary based on sampling error due to different training and test sets.
  - Bummer… what should we do?
- Average results over multiple training and test sets (splits of the overall data) for the best results.
  - Bummer… that means we need ***lots*** of labeled data…

12

N-Fold Cross-Validation

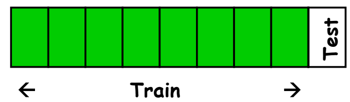
- Ideally: test, training sets are independent on each trial.
  - But this would require too much labeled data.
- Cool idea:
  - Partition data into *N* equal-sized disjoint segments.
  - Run *N* trials, each time hold back a different segment for testing
  - Train on the remaining *N*–1 segments.
- This way, at least test-sets are independent.
- Report average classification accuracy over the *N* trials.
- Typically, *N* = 10.

Also nice to report standard deviation of averages

13

Cross validation

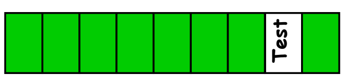
- Partition examples into *k* disjoint equiv classes
- Now create *k* training sets
  - Each set is union of all equiv classes ***except one***
  - So each set has (k-1)/k of the original training data



14

Cross Validation

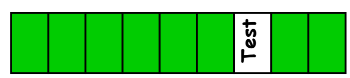
- Partition examples into *k* disjoint equiv classes
- Now create *k* training sets
  - Each set is union of all equiv classes ***except one***
  - So each set has (k-1)/k of the original training data



15

Cross Validation

- Partition examples into *k* disjoint equiv classes
- Now create *k* training sets
  - Each set is union of all equiv classes ***except one***
  - So each set has (k-1)/k of the original training data



16

Learning Curves

- In practice, labeled data is usually rare and expensive.
  - So….would like to know how performance varies with the number of training instances.
- *Learning curves* plot classification accuracy on independent test data (*Y* axis) versus number of training examples (*X* axis).

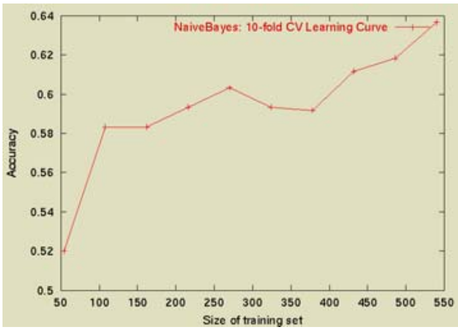
17

N-Fold Learning Curves

- Want learning curves averaged over multiple trials.
- Use *N*-fold cross validation to generate *N* full training and test sets.
- For each trial,
  - Train on increasing fractions of the training set,
  - Measure accuracy on test data for each point on the desired learning curve.

18

Sample Learning Curve (Yahoo Science Data)



19

Co-Training Motivation

- Learning methods need labeled data
  - Lots of <x, f(x)> pairs
  - Hard to get… (who wants to label data?)
- But unlabeled data is usually plentiful…
  - Could we use this instead??????

20

Co-training

Small labeled data needed

- Suppose each instance has two parts:

  x = [x1, x2]

  x1, x2 conditionally independent given f(x)
- Suppose each half can be used to classify instance

  ∃f1, f2 such that f1(x1) = f2(x2) = f(x)
- Suppose f1, f2 are learnable

  f1 ∈ H1, f2 ∈ H2, ∃ learning algorithms A1, A2

[x1, x2] —A1→ <[x1, x2], f1(x1)> —A2→ $\tilde{f2}$

Unlabeled Instances — Labeled Instances — Hypothesis

21

Observations

- Can apply A1 to generate as much training data as one wants
  - If x1 is conditionally independent of x2 / f(x),
  - then the error in the labels produced by A1
  - ***will look like random noise to A2 !!!***
- Thus no limit to quality of the hypothesis A2 can make

22

It really works!

- Learning to classify web pages as course pages
  - x1 = bag of words on a page
  - x2 = bag of words from all anchors pointing to a page
- Naïve Bayes classifiers
  - 12 labeled pages
  - 1039 unlabeled

	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

Table 2: Error rate in percent for classifying web pages as course home pages. The top row shows errors when training on only the labeled examples. Bottom row shows errors when co-training, using both labeled and unlabeled examples.

23

Focussed Crawling

- Cho paper
  - Looks at heuristics for managing URL queue
  - Aim1: completeness
  - Aim2: just topic pages
- Prioritize if word in anchor / URL
- Heuristics:
  - Pagerank
  - #backlinks

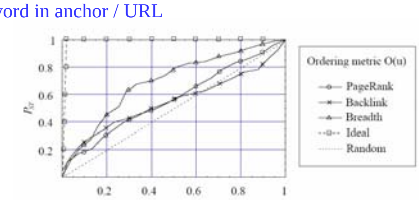


Figure 12: Basic similarity-based crawler. $I(p) = IS(p)$; topic is *computer*.

24

Modified Algorithm

- Page is hot if:
  - Contains keyword in title, or
  - Contains 10 instances of keyword in body, or
  - Distance(page, hot-page) < 3

25

Results

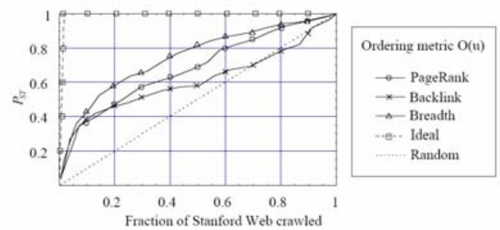


Figure 14: Modified similarity-based crawler. $I(p) = IS(p)$; topic is *computer*.

26

More Results

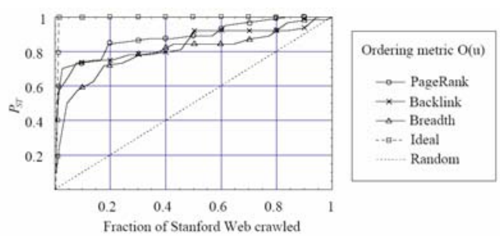
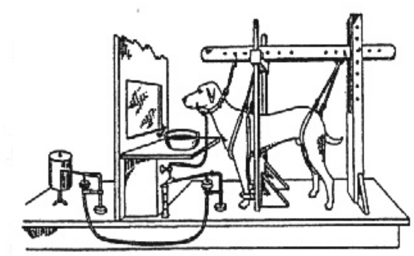


Figure 15: Modified similarity-based crawler. Topic is *admission*.

27

Reinforcement Learning



Ever Feel Like Pavlov's Poor Dog?

--

How is learning to act possible when…

- Actions have non-deterministic effects
  - Which are initially unknown
- Rewards / punishments are infrequent
  - Often at the end of long sequences of actions
- Learner must decide what actions to take
- World is large and complex

29

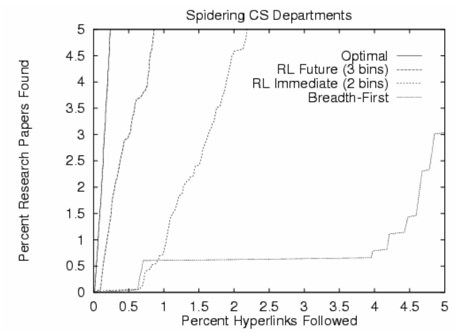
Applications to the Web
Focused Crawling

- Limited resources
  - Fetch most ***important*** pages first
- Topic specific search engines
  - Only want pages which are ***relevant*** to topic
- Minimize stale pages
  - Efficient re-fetch to keep index timely
  - How track the rate of change for pages?

30

Performance

Rennie & McCallum (ICML-99)



31

Information Extraction

"The Truth Is Out There"

Papers: Scaling Question Answering for the Web (WWW '01)
Web-Scale Information Extraction in KnowItAll (WWW '04)

32

Information Goals

Finding Topics
Where can I find pages about skiing?

vs.

Finding Answers
Who killed Lincoln? "John Wilkes Booth"

33

Mulder

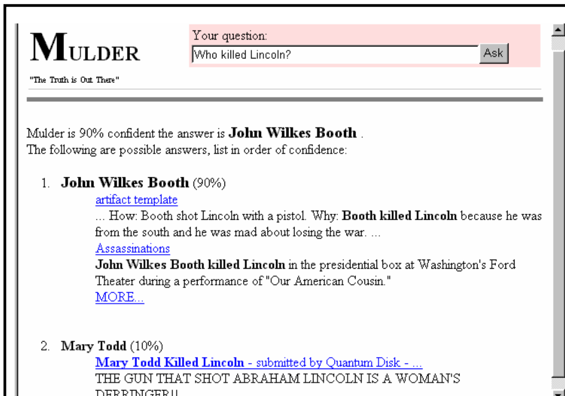
- Question Answering System
  - User asks Natural Language question: "Who killed Lincoln?"
  - Mulder answers: "John Wilkes Booth"
- KB = Web/Search Engines
- Domain-independent
- Fully automated

34

Mulder versus…

	Web Coverage	Direct Answers	Automated	Ease of use
Mulder	Wide	Yes	Yes	Easy
Directories	Narrow	No	No	Easy
Search Engines	Wide	No	Yes	Difficult
AskJeeves	Narrow	No	No	Easy

35


MULDER
"The Truth is Out There"

Your question: Who killed Lincoln? Ask

Mulder is 90% confident the answer is **John Wilkes Booth** .
The following are possible answers, list in order of confidence:

1. **John Wilkes Booth** (90%)
artifact template
... How: Booth shot Lincoln with a pistol. Why: **Booth killed Lincoln** because he was from the south and he was mad about losing the war. ...
Assassinations
John Wilkes Booth killed Lincoln in the presidential box at Washington's Ford Theater during a performance of "Our American Cousin."
MORE...

2. **Mary Todd** (10%)
Mary Todd Killed Lincoln - submitted by Quantum Disk - ...
THE GUN THAT SHOT ABRAHAM LINCOLN IS A WOMAN'S DERRINGER!!

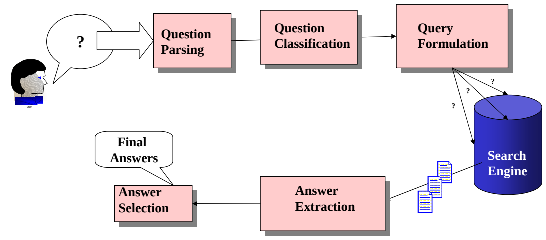
36

Challenges

- Web: Huge
  - Difficult to pinpoint facts
- Noise
  - "Conspiracy theorists believes that Mary Todd killed Lincoln"
- False claims
  - "John Glenn is the first American in space"

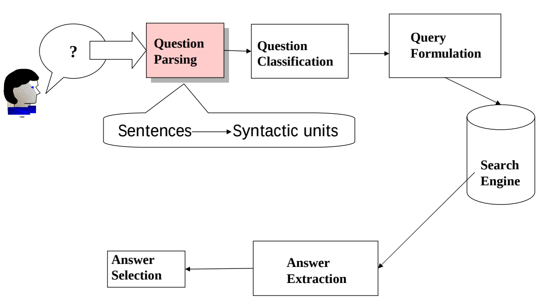
37

Architecture



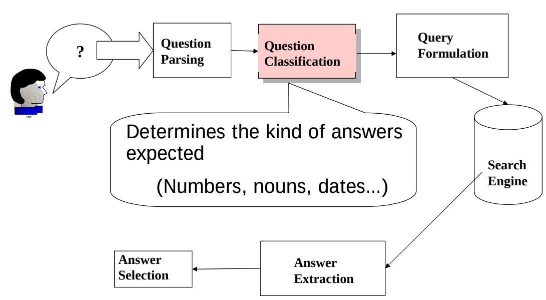
38

Natural Language Parsing



39

Question Classification



40

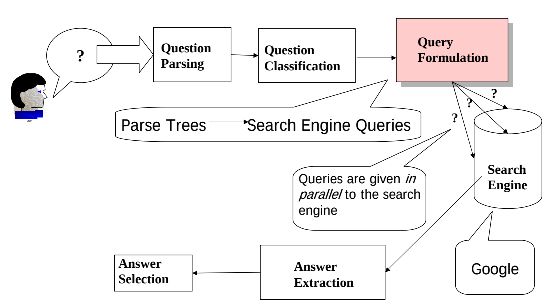
Question Classification

Rule-based system

- Question Words (who, when)
- Question subject (what *height*)
  - LinkParser [Sleator et al., 91]
    - Recovers relationships among words in a sentence
  - WordNet [Miller 90]
    - Semantic network: relationships between words
    - Subtyping: height – magnitude – number

41

Query Formulation



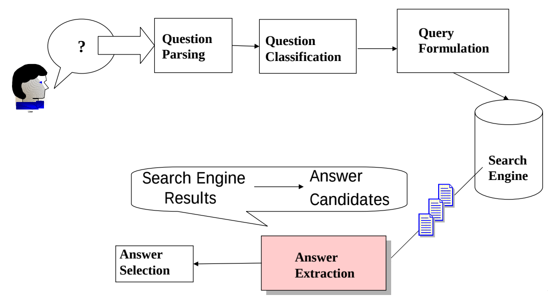
42

Query Formulation

- Transformation examples
  - Grammatical
    "Lincoln was killed by *person*" "*person* killed Lincoln"
  - Query Expansion: "*person* murdered Lincoln" "*person* assassinated Lincoln"…
  - Verb Conversion: "When did Lincoln die?" "Lincoln died *in/on date*"

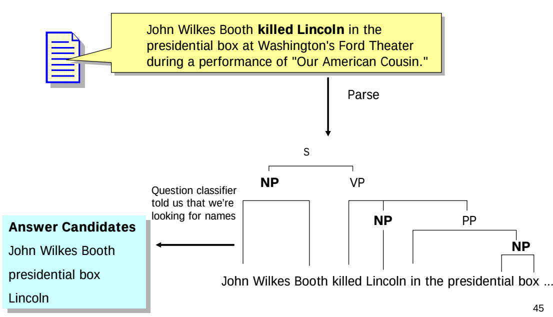
43

Answer Extraction



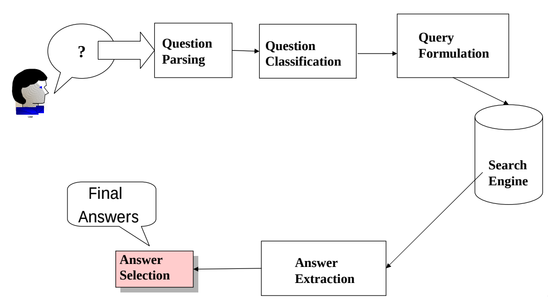
44

Answer Extraction



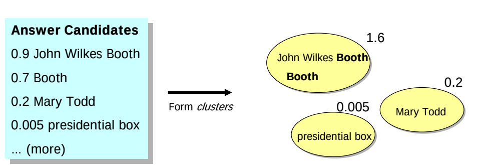
45

Answer Selection



46

Answer Selection



- Score and select top candidates
  - Proximity with query words
  - Hints from query formulation

- Clustering – grouping phrases with common words
  - Reduces answer variations
  - Reduces noise – *Assume the truth prevails over others*

47

Empirical Evaluations

- Test Suite
  - NIST's TREC-8 (The 8th Text REtrieval Conference)
  - ~200 questions
  - Not guaranteed to find answers on the web
- What experiments would you run?
  - Contributions of each Mulder module
  - Mulder VS. Google VS. AskJeeves ???

48

Experimental Methodology

- Idea: In order to answer n questions, how much *user effort* has to be exerted
- Implementation:
  - A question is answered if
    - the answer phrases are found in the result pages returned by the service, or
    - they are found in the web pages pointed to by the results.
  - Bias in favor of Mulder's opponents

49

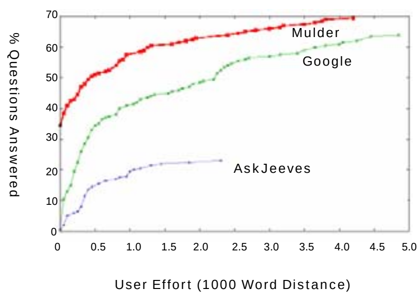
Experimental Methodology

- User Effort = Word Distance
  - # of words read before answers are encountered

- Google/AskJeeves – query with the original question

50

Comparison Results



51

Contributions of Modules

- Compare Mulder with stripped down variants.

System	Total effort / Total Effort Mulder
Mulder	1.0
No Answer Selection	2.3
No Query Formulation	3.0
No Answer Extraction	3.8
Nothing but Google	6.6

52

KnowItAll

- Mulder on Steriods.
- Instead of answering one question --- collect millions and millions of facts.
- How can we do this?

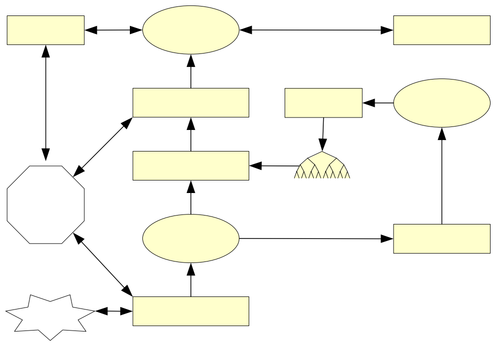
53

KnowItAll Architecture

- Extraction engine: rules for extracting information from text.
- Assessor: uses PMI-IR to assess probability that extractions are correct.
- Rule Learner: automatically learn new extraction rules.

54

KnowItAll System Architecture



Example Extraction Rule

A rule template for instanceOf(Class1):

```
    NP1 "such as" NP2
    & head(NP1)= label(Class1) &
    properNoun(head(NP2))
 =>
    instanceOf(Class1, head(NP2))
```

```
Example: High quality laptops such as the
 Thinkpad T-40.
```

Yields: instanceOf(laptops, Thinkpad T-40).

Web-scale Validation of Facts

Probability of fact ϕ, given evidence f_1, f_2, f_n using Bayes rule with independence assumption.

$$P(\phi \mid f_1, f_2, \ldots f_n) = \frac{P(\phi)\Pi_i P(f_i \mid \phi)}{Z}$$

$$Z = P(\phi)\Pi_i P(f_i \mid \phi) + P(\neg\phi)\Pi_i P(f_i \mid \neg\phi)$$

Pointwise mutual information of instance I with discriminator phrase D, based on search engine hit counts

$$pmiScore(I, D) = \frac{|Hits(I + D)|}{|Hits(I)|}$$

Ontolo
Extenc

Features for Web-scale Validation

Features are based on PMI score thresholds.

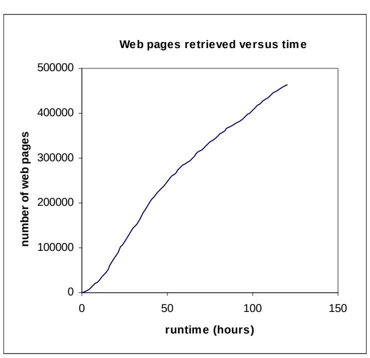
Find threshold that best separates positive from negative training instances (maximize entropy).

$$f_i = pmiScore \geq \tau_i$$

Estimate the probability of score over threshold, given that the instance is positive (or negative). Probability is the proportion of a holdout set H with score over threshold, with m-smoothing.

$$P(pmiScore \geq \tau_i \mid \phi) = \frac{|H_{\tau_i}| + P(\phi)m}{|H| + m}$$

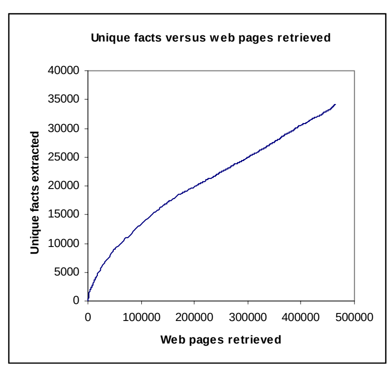
C



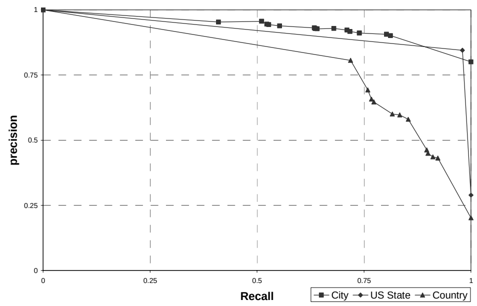
Searc
Engin

Extractor

Extract



WWW



61

KnowItAll System Performance

- Longest run to date: 5 days
  - Substantially longer runs are possible
- Performance of current prototype:
  - 3 web pages examined per second
  - 0.9 sentences containing extractions found per second
  - Web-scale validation assigns probability > .80 to nearly half of the extractions. These are facts.
  - KnowItAll achieve precision of 95% for facts.

62

Conclusion

You ain't seen nuffin' yet!

63