Workshop 2: Evaluation

Bill MacCartney
CS224U
21 February 2013

Why does evaluation matter?

In your final project, you will have:

- Identified a problem
- Explained why the problem matters
- Examined existing solutions, and found them wanting
- Proposed a new solution, and described its implementation

So the key question will be:

- Did you solve the problem?

The answer need not be yes, but the question must be addressed!

Who is it for?

Evaluation matters for many reasons, and for multiple parties:

- For future researchers
  - Should I adopt the methods used in this paper?
  - Is there an opportunity for further gains in this area?

- For reviewers
  - Does this paper make a useful contribution to the field?

- For yourself
  - Should I use method/data/classifier/... A or B?
  - What's the optimal value for parameter X?
  - What features should I add to my feature representation?
  - How should I allocate my remaining time and energy?

The role of data in evaluation

- Evaluations should be *empirical* — i.e., data-driven
- We are scientists!
  - Well, or engineers — either way, we're empiricists!
  - Not some hippie tree-hugging philosophers or poets
- You're trying to solve a real problem
  - Need to verify that your solution solves real problem instances
- So evaluate the output of your system on real inputs
  - Realistic data, not toy data or artificial data
  - Ideally, plenty of it

Agenda

1. Introduction
2. Kinds of evaluation
3. Data management
4. Evaluation metrics
5. Comparative evaluations
6. Other aspects of evaluation
7. Conclusion

Kinds of evaluation

Quantitative	vs.	Qualitative
Automatic	vs.	Manual
Intrinsic	vs.	Extrinsic
Formative	vs.	Summative

Quantitative vs. qualitative

- Quantitative evaluations should be primary
  - Evaluation metrics — *much* more below
  - Tables & graphs & charts, oh my!

- But qualitative evaluations are useful too!
  - Examples of system outputs
    - A tremendous aid to your readers' understanding
  - Error analysis
    - Can drive system development (e.g. feature engineering)
    - Again, a tremendous aid to your readers' understanding
  - Interactive demos
    - A great way to gain visibility and impact for your work
    - Example: the ReVerb demo

Examples of system outputs

brief (noun): affidavit 0.13, petition 0.05, memorandum 0.05, motion 0.05, lawsuit 0.05, deposition 0.05, slight 0.05, prospectus 0.04, document 0.04, paper 0.04, …

brief (verb): tell 0.09, urge 0.07, ask 0.07, meet 0.06, appoint 0.06, elect 0.05, name 0.05, empower 0.05, summon 0.05, overrule 0.04, …

brief (adjective): lengthy 0.13, short 0.12, recent 0.09, prolonged 0.09, long 0.09, extended 0.09, daylong 0.08, scheduled 0.08, stormy 0.07, planned 0.06, …

from Lin 1998

Examples of system outputs

n	Most negative n-grams	Most positive n-grams
1	bad; boring; dull; flat; pointless; tv; neither; pretentious; badly; worst; lame; mediocre; lack; routine; loud; bore; barely; stupid; tired; poorly; suffers; heavy;nor; choppy; superficial	touching; enjoyable; powerful; warm; moving; culture; flaws; provides; engrossing; wonderful; beautiful; quiet; socio-political; thoughtful; portrait; refreshingly; chilling; rich; beautifully; solid;
2	how bad; by bad; dull .; for bad; to bad; boring .; , dull; are bad; that bad; boring ,; , flat; pointless .; badly by; on tv; so routine; lack the; mediocre .; a generic; stupid ,; abysmally pathetic	the beautiful; moving,; thoughtful and; , inventive; solid and; a beautiful; a beautifully; and hilarious; with dazzling; provides the; provides.; and inventive; as powerful; moving and; a moving; a powerful
3	. too bad; exactly how bad; and never dull; shot but dull; is more boring; to the dull; dull, UNK; it is bad; or just plain; by turns pretentious; manipulative and contrived; bag of stale; is a bad; the whole mildly; contrived pastiche of; from this choppy; stale material.	funny and touching; a small gem; with a moving; cuts, fast; , fine music; smart and taut; culture into a; romantic , riveting; ... a solid; beautifully acted .; , gradually reveals; with the chilling; cast of solid; has a solid; spare yet audacious; ... a polished; both the beauty;
5	boring than anything else.; a major waste ... generic; nothing i hadn't already; ,UNK plotting;superficial; problem ? no laughs.; ,just horribly mediocre .; dull, UNK feel.; there's nothing exactly wrong; movie is about a boring; essentially a collection of bits	reminded us that a feel-good; engrossing, seldom UNK,; between realistic characters showing honest; a solid piece of journalistic; easily the most thoughtful fictional; cute, funny, heartwarming; with wry humor and genuine; engrossing and ultimately tragic.;
8	loud, silly, stupid and pointless.; dull, dumb and derivative horror film.; UNK's film, a boring, pretentious; this film biggest problem ? no laughs.; film in the series looks and feels tired; do draw easy chuckles but lead nowhere.; stupid, infantile, redundant, sloppy	shot in rich , shadowy black-and-white , devils an escapist confection that 's pure entertainment .; , deeply absorbing piece that works as a; ... one of the most ingenious and entertaining; film is a riveting , brisk delight .; bringing richer meaning to the story 's;

from Socher et al. 2011

Automatic vs. manual evaluation

- Automatic evaluation
  - Typically: compare system outputs to some “gold standard”
  - Pro: cheap, fast
  - Pro: objective, reproducible
  - Con: may not reflect end-user quality
  - Especially useful during development (formative evaluation)
- Manual evaluation
  - Generate system outputs, have humans assess them
  - Pro: directly assesses real-world utility
  - Con: expensive, slow
  - Con: subjective, inconsistent
  - Most useful in final assessment (summative evaluation)

Automatic evaluation

- Automatic evaluation against human-annotated data
  - But human-annotated data is not available for many tasks
  - Even when it is, quantities are often rather limited
- Automatic evaluation against synthetic data
  - Example: pseudowords (*bananadoor*) in WSD
  - Example: cloze (completion) experiments
    - Chambers & Jurafsky 2008; Busch, Colgrove, & Neidert 2012
  - Pro: virtually infinite quantities of data
  - Con: lack of realism

With a pile of browning bananadoors, I …

… like a bananadoor to another world …

… highland bananadoors are a vital crop …

… how to construct a sliding bananadoor.

Known events:
(pleaded subj), (admits subj), (convicted obj)

Likely Events:

sentenced obj	0.89	indicted obj	0.74
paroled obj	0.76	fined obj	0.73
fired obj	0.75	denied subj	0.73

Manual evaluation

- Generate system outputs, have humans evaluate them
- Pros: direct assessment of real-world utility
- Cons: expensive, slow, subjective, inconsistent
- But sometimes unavoidable! (Why?)
- Example: cluster intrusion in Yao et al. 2012
- Example: Banko et al. 2008

Intrinsic vs. extrinsic evaluation

- Intrinsic (*in vitro*, task-independent) evaluation
  - Compare system outputs to some ground truth or gold standard
- Extrinsic (*in vivo*, task-based, end-to-end) evaluation
  - Evaluate impact on performance of a larger system of which your model is a component
  - Pushes the problem back — need way to evaluate larger system
  - Pro: a more direct assessment of “real-world” quality
  - Con: often very cumbersome and time-consuming
  - Con: real gains may not be reflected in extrinsic evaluation
- Example from automatic summarization
  - Intrinsic: do summaries resemble human-generated summaries?
  - Extrinsic: do summaries help humans gather facts quicker?

Formative vs. summative evaluation

When the cook tastes the soup, that's formative;
when the customer tastes the soup, that's summative.

- Formative evaluation: guiding further investigations
  - Typically: lightweight, automatic, intrinsic
  - Compare design option A to option B
  - Tune parameters: smoothing, weighting, learning rate
- Summative evaluation: reporting results
  - Compare your approach to previous approaches
  - Compare different variants of your approach

Agenda

1. Introduction
2. Kinds of evaluation
3. Data management
4. Evaluation metrics
5. Comparative evaluations
6. Other aspects of evaluation
7. Conclusion

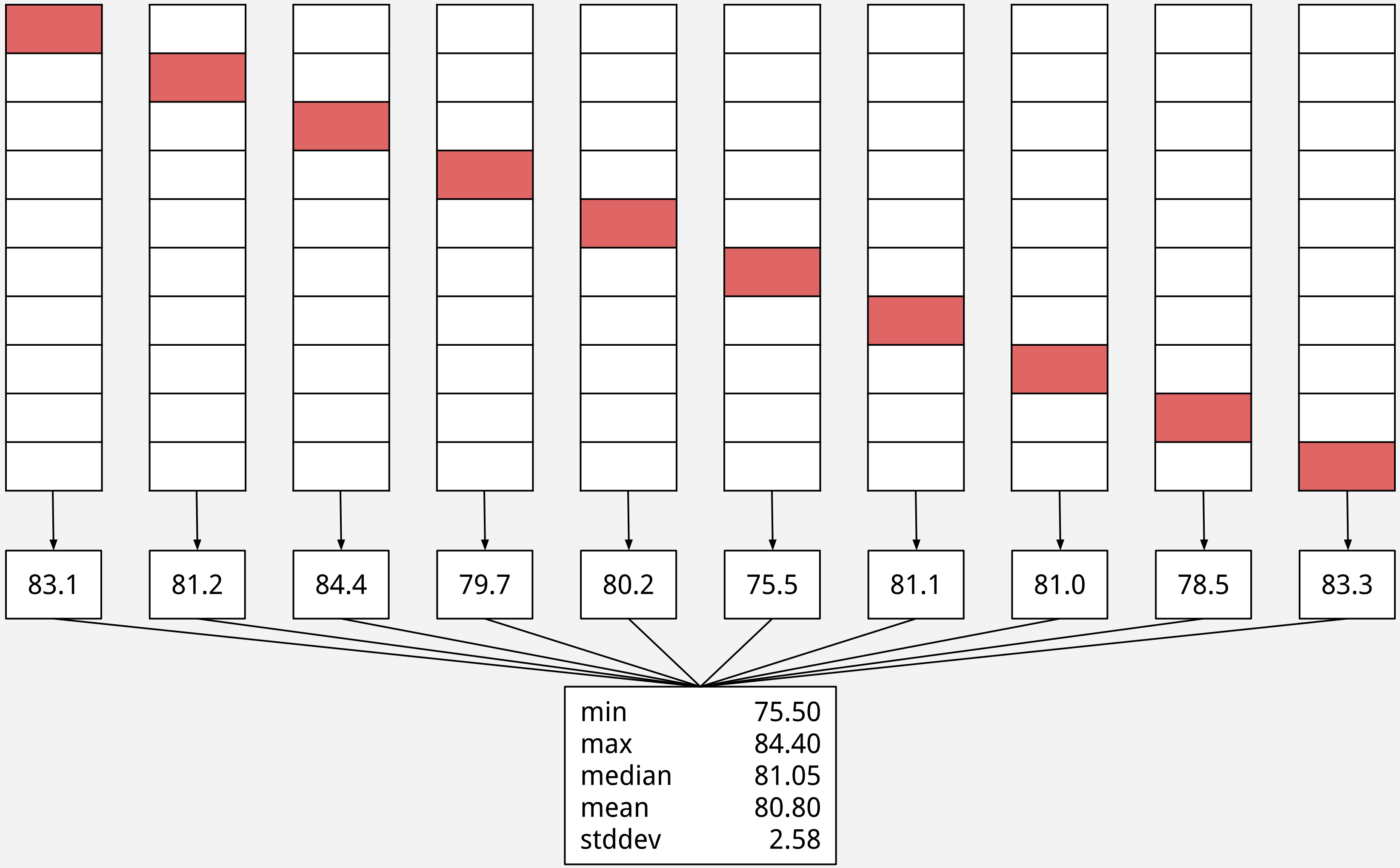
The train/test split

- Evaluations on training data overestimate real performance!
  - Need to test model's ability to *generalize*, not just memorize
  - But testing on training data can still be useful — how?
- So, sequester test data, use *only* for summative evaluation
  - Typically, set aside 10% or 20% of all data for final test set
  - Don't peek!
- Beware of subtle ways that test data can get tainted
  - Using same test data in repeated experiments
  - "Community overfitting", e.g. on PTB parsing
  - E.g., matching items to users: partition on *users*, not matches

Development data

- Also known as “devtest” or “validation” data
- Used as test data during formative evaluations
  - Keep *real* test data pure until summative evaluation
- Useful for selecting (discrete) design options
  - Which categories of features to activate
  - Choice of classification (or clustering) algorithm
  - VSMs: choice of distance metric, normalization method, ...
- Useful for tuning (continuous) hyperparameters
  - Smoothing / regularization parameters
  - Combination weights in ensemble systems
  - Learning rates, search parameters

10-fold cross-validation (10CV)



k-fold cross-validation

- Pros
  - Make better use of limited data
  - Less vulnerable to quirks of train/test split
  - Can estimate variance (etc.) of results
  - Enables crude assessment of statistical significance
- Cons
  - Slower (in proportion to k)
  - Doesn't keep test data "pure" (if used in development)
- LOOCV = leave-one-out cross-validation
  - Increase k to the limit: the total number of instances
  - Magnifies both pros and cons

Agenda

1. Introduction
2. Kinds of evaluation
3. Data management
4. Evaluation metrics
5. Comparative evaluations
6. Other aspects of evaluation
7. Conclusion

Evaluation metrics

- An evaluation metric is a function: model × data $\rightarrow \mathbb{R}$
- Can involve both manual and automatic elements
- Can serve as an objective function during development
    - For formative evaluations, identify *one* metric as primary
    - Known as “figure of merit”
    - Use it to guide design choices, tune hyperparameters
- You may use standard metrics, or design your own
    - Using standard metrics facilitates comparisons to prior work
    - But new problems may require new evaluation metrics
    - Either way, have good *reasons* for your choice

Example: evaluation metrics

Evaluation metrics are the *columns* of your main results table:

System	Pairwise				B^3		
	Prec.	Rec.	F-0.5	MCC	Prec.	Rec.	F-0.5
Rel-LDA/300	0.593	0.077	0.254	0.191	0.558	0.183	0.396
Rel-LDA/1000	0.638	0.061	0.220	0.177	0.626	0.160	0.396
HAC	0.567	0.152	0.367	0.261	0.523	**0.248**	0.428
Local	0.625	0.136	0.364	0.264	0.626	0.225	0.462
Local+Type	0.718	0.115	0.350	0.265	**0.704**	0.201	0.469
Our Approach	**0.736**	**0.156**	**0.422**	**0.314**	0.677	0.233	**0.490**
Our Approach+Type	0.682	0.110	0.334	0.250	0.687	0.199	0.460

from Yao et al. 2012

Evaluation metrics for classification

- Contingency tables & confusion matrices
- Accuracy
- Precision & recall
- F-measure
- AUC (area under ROC curve)
- Sensitivity & specificity
- PPV & NPV (positive/negative predictive value)
- MCC (Matthews correlation coefficient)

Contingency tables

- In binary classification, each instance has actual label ("gold")
- The model assigns to each instance a predicted label ("guess")
- A pair of labels [actual, predicted] determines an outcome
  - E.g., [actual:false, predicted:true] → false positive (FP)
- The contingency table counts the outcomes
- Forms basis of many evaluation metrics: accuracy, P/R, MCC, ...

		guess	
		false	true
gold	false	TN true negative	FP false positive
	true	FN false negative	TP true positive

		guess	
		false	true
gold	false	51	9
	true	4	36

Confusion matrices

- Generalizes the contingency table to multiclass classification
- Correct predictions lie on the main diagonal
- Large off-diagonal counts reveal interesting "confusions"

		guess Y	guess N	guess U	
gold	Y	67	4	31	102
	N	1	16	4	21
	U	7	7	46	60
		75	27	81	183

from MacCartney & Manning 2008

Accuracy

- Accuracy: percent correct among *all* instances
- The most basic and ubiquitous evaluation metric
- But, it has serious limitations (what?)

gold \ guess	F	T	
F	86	2	88
T	9	3	12
	95	5	100

$$\text{accuracy} = \frac{86 + 3}{100} = 89.0\%$$

gold \ guess	Y	N	U	
Y	67	4	31	102
N	1	16	4	21
U	7	7	46	60
	75	27	81	183

$$\text{accuracy} = \frac{67 + 16 + 46}{183} = 70.5\%$$

Precision & recall

- Precision: % correct among items where guess=true
- Recall: % correct among items where gold=true
- Preferred to accuracy, especially for highly-skewed problems

guess

		F	T	
gold	F	86	2	88
	T	9	3	12
		95	5	100

$$\text{recall} = \frac{3}{12} = 25.0\%$$

$$\text{precision} = \frac{3}{5} = 60.0\%$$

F_1

- It's helpful to have a single measure which combines P and R
- But we *don't* use the arithmetic mean of P and R (why not?)
- Rather, we use the harmonic mean: $F_1 = 2PR / (P + R)$

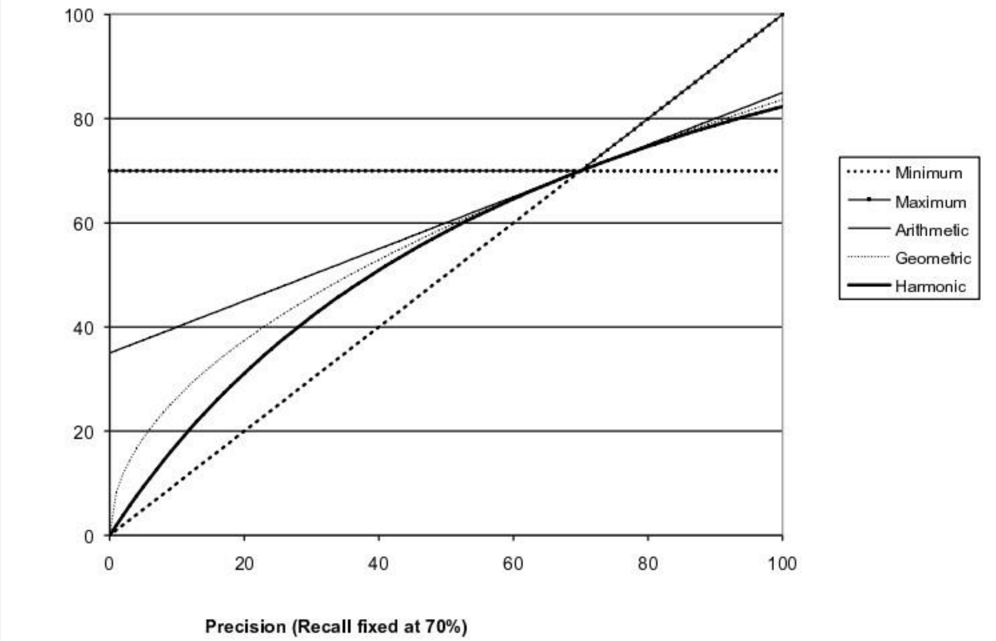
		guess		
		F	T	
gold	F	86	2	88
	T	9	3	12
		95	5	100

$$\text{recall} = \frac{3}{12} = 25.0\%$$

$$\text{precision} = \frac{3}{5} = 60.0\%$$

$$F_1 = 35.3\%$$

Why use harmonic mean?



► **Figure 8.1** Graph comparing the harmonic mean to other means. The graph shows a slice through the calculation of various means of precision and recall for the fixed recall value of 70%. The harmonic mean is always less than either the arithmetic or geometric mean, and often quite close to the minimum of the two numbers. When the precision is also 70%, all the measures coincide.

from [Manning et al. 2009]

F-measure

- Some applications need more precision; others, more recall
- F_β is the *weighted* harmonic mean of P and R
- $F_\beta = (1 + \beta^2)PR / (\beta^2 P + R)$

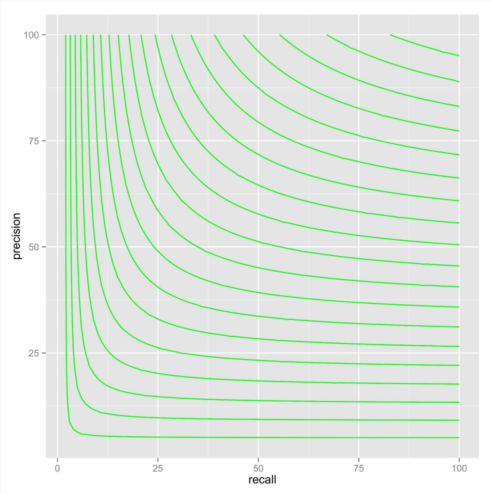
β = 2.0 (favor recall)
β = 1.0 (neutral)
β = 0.5 (favor precision)

precision \ recall	0.10	0.30	0.60	0.90
0.10	0.10 0.10 0.10	0.21 0.15 0.12	0.30 0.17 0.12	0.35 0.18 0.12
0.30	0.12 0.15 0.21	0.30 0.30 0.30	0.50 0.40 0.33	0.64 0.45 0.35
0.60	0.12 0.17 0.30	0.33 0.40 0.50	0.60 0.60 0.60	0.82 0.72 0.64
0.90	0.12 0.18 0.35	0.35 0.45 0.64	0.64 0.72 0.82	0.90 0.90 0.90

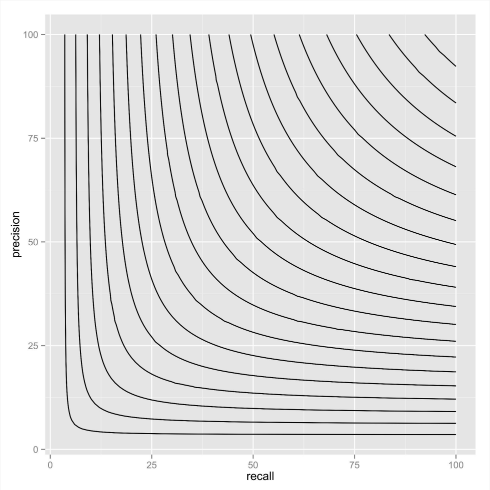
F-measure

- Some applications need more precision; others, more recall
- F_β is the *weighted* harmonic mean of P and R
- $F_\beta = (1 + \beta^2)PR / (\beta^2 P + R)$

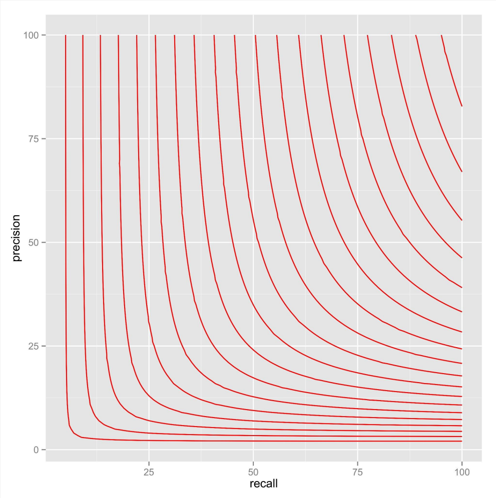
β = 0.5 (favor precision)



β = 1.0 (neutral)

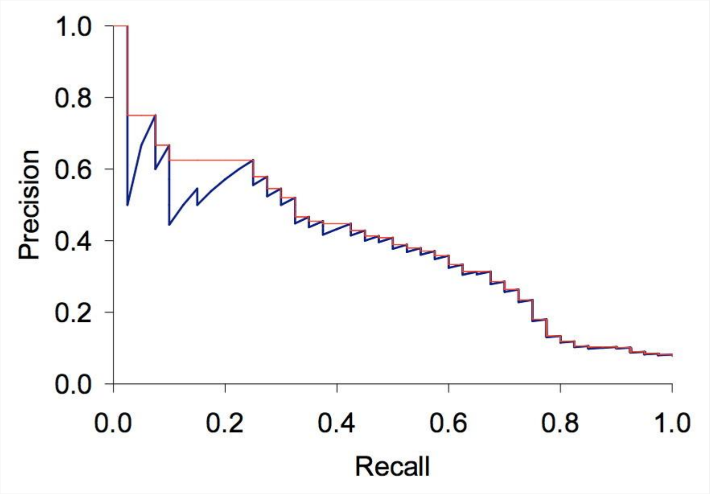


β = 2.0 (favor recall)



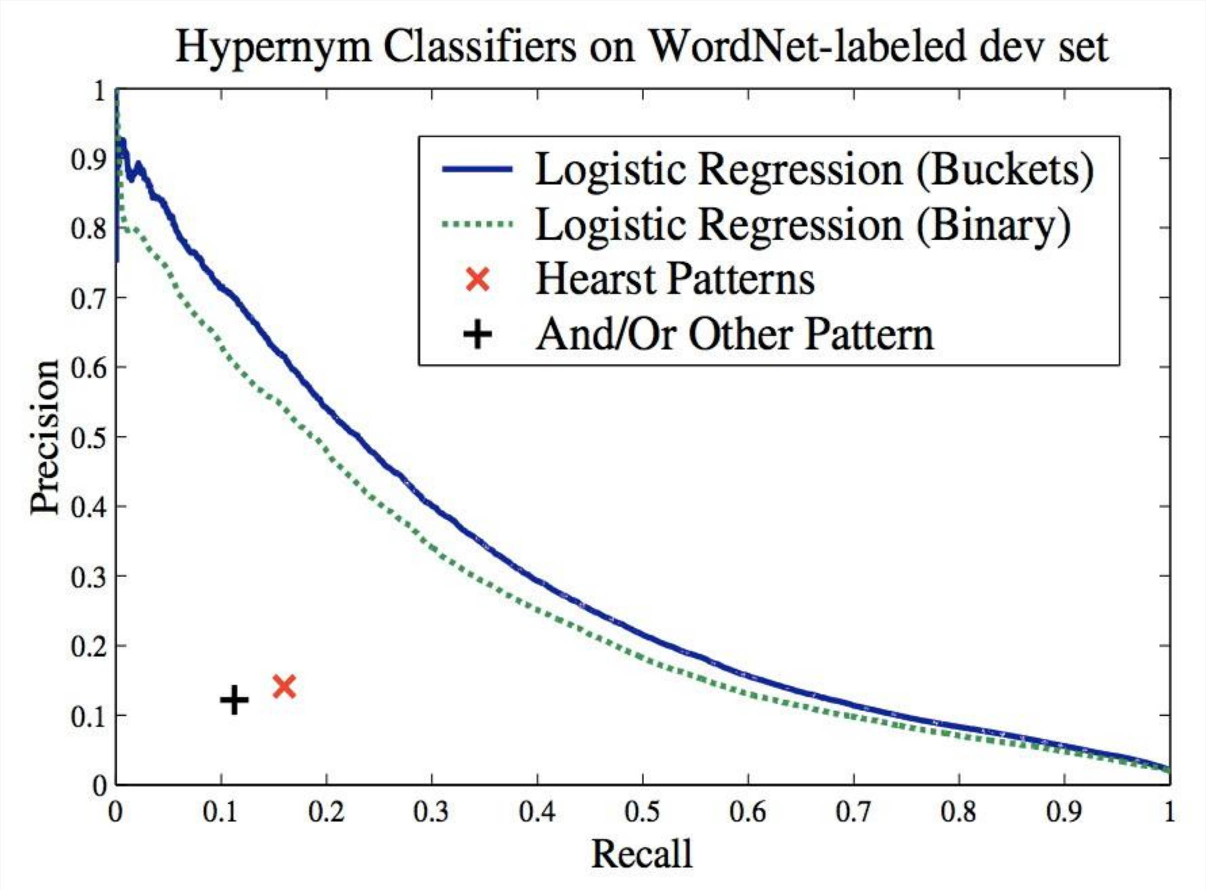
Precision vs. recall

- Typically, there's a trade-off between precision and recall
  - High threshold → high precision, low recall
  - Low threshold → low precision, high recall
- P/R curve facilitates making an explicit choice on trade-off
- Always put recall on x-axis, and expect noise on left (why?)



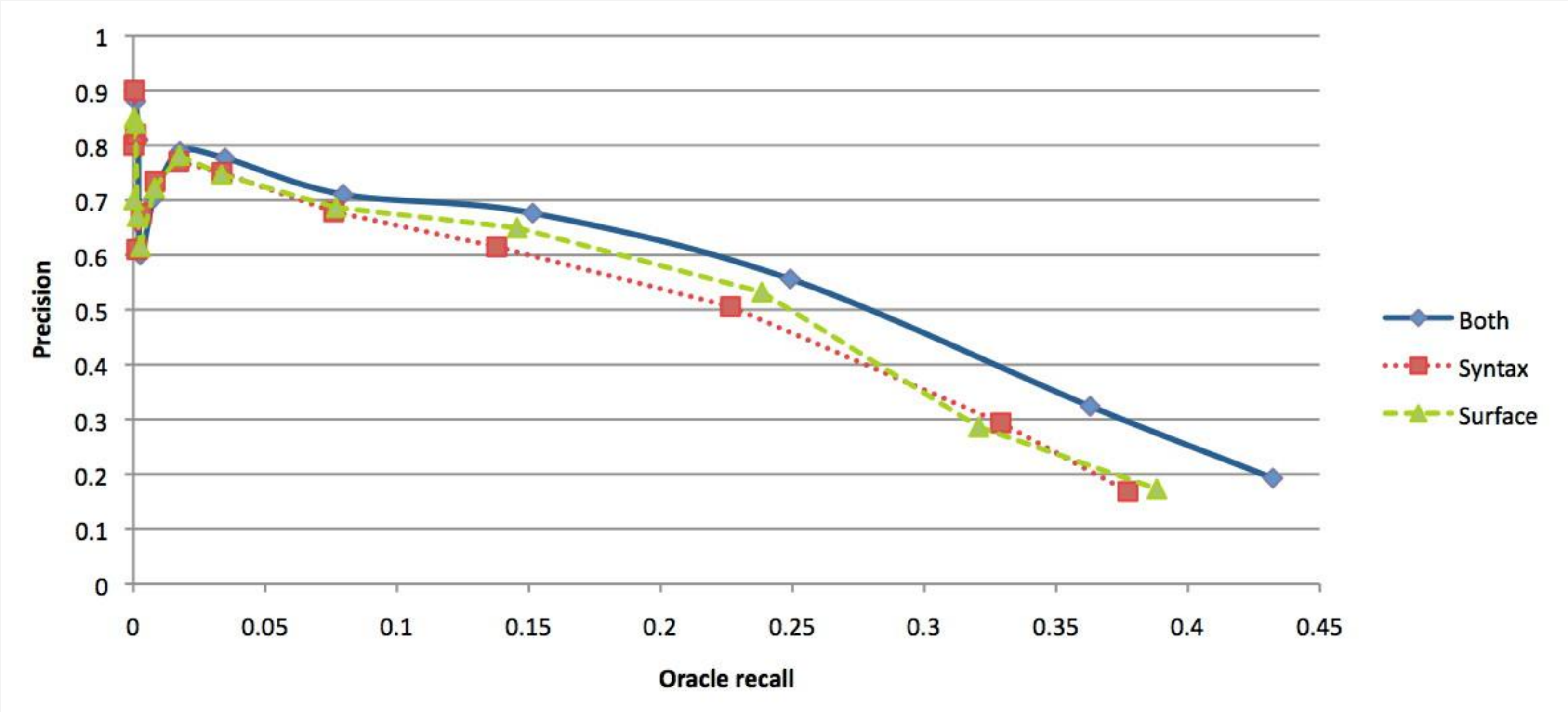
from [Manning et al. 2009]

Precision/recall curve example



from [Snow et al. 2005]

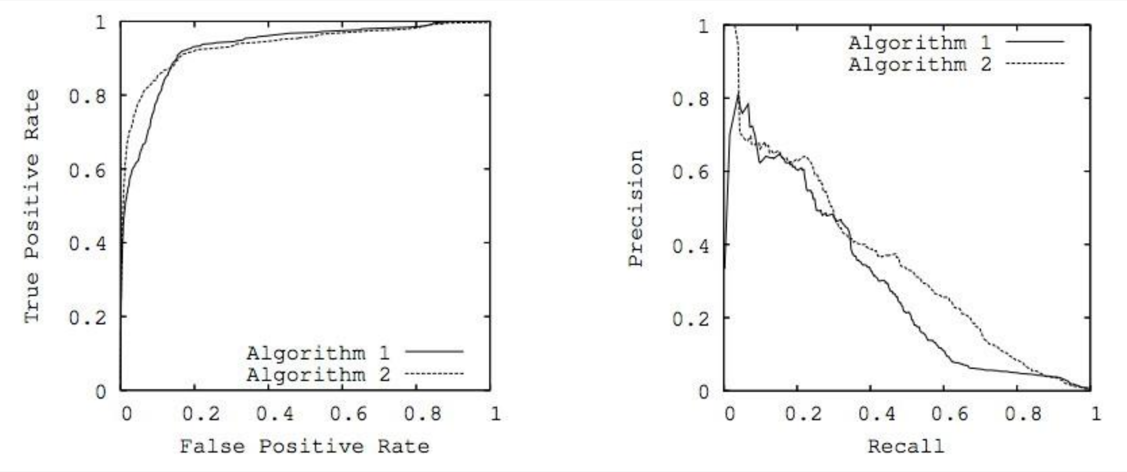
Precision/recall curve example



from Mintz et al. 2009

ROC curves and AUC

- ROC curve = receiver operating characteristic curve
  - An alternative to P/R curve used in other fields (esp. EE)
- AUC = area under (ROC) curve
  - Like F1, a single metric which promotes both P and R
  - But doesn't permit specifying tradeoff, and generally *unreliable*



from Davis & Goadrich 2006

Sensitivity & specificity

- Sensitivity & specificity look at % correct by actual label
  - Sensitivity: % correct among items where gold=true (= recall)
  - Specificity: % correct among items where gold=false
- An alternative to precision & recall
  - More common in statistics literature

		guess F	guess T	
gold	F	86	2	88
	T	9	3	12
		95	5	100

$$\text{specificity} = \frac{86}{88} = 97.7\%$$

$$\text{sensitivity} = \frac{3}{12} = 25.0\%$$

PPV & NPV

- PPV & NPV look at % correct by predicted label
  - PPV: % correct among items where guess=true (= precision)
  - NPV: % correct among items where guess=false
- An alternative to precision & recall
  - More common in statistics literature

		guess F	guess T	
gold	F	86	2	88
	T	9	3	12
		95	5	100

$$\text{NPV} = \frac{86}{95} = 90.5\% \qquad \text{PPV} = \frac{3}{5} = 60.0\%$$

Matthews correlation coefficient (MCC)

- Correlation between actual & predicted classifications
- Random guessing yields 0; perfect prediction yields 1

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

MCC (precision \ recall)	0.05	0.35	0.65	0.95
0.05	-0.90	—	—	—
0.35	-0.11	-0.30	—	—
0.65	0.08	0.22	0.30	0.36
0.95	0.21	0.55	0.74	0.90

with prevalence = 0.50

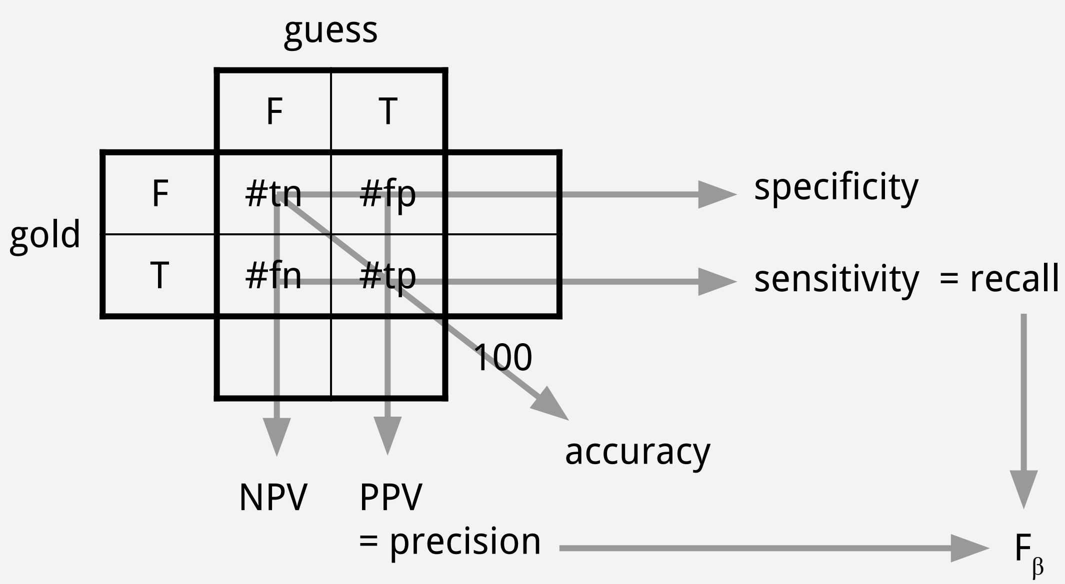
MCC (precision \ recall)	0.05	0.35	0.65	0.95
0.05	-0.06	-0.15	—	—
0.35	0.10	0.28	0.38	0.76
0.65	0.17	0.45	0.61	0.74
0.95	0.22	0.57	0.78	0.94

with prevalence = 0.10

Recap: metrics for classifiers

accuracy	proportion of all items predicted correctly
error	proportion of all items predicted incorrectly
sensitivity	accuracy over items actually true
specificity	accuracy over items actually false
PPV	accuracy over items predicted true
NPV	accuracy over items predicted false
precision	accuracy over items predicted true
recall	accuracy over items actually true
F1	harmonic mean of precision and recall
MCC	correlation between actual & predicted classifications

Recap: metrics for classifiers



Multiclass classification

- Precision, recall, F_1, MCC, ... are for binary classification
- For multiclass classification, compute these stats *per class*
  - For each class, project into binary classification problem
  - TRUE = this class; FALSE = all other classes
- Then average the results
  - Macro-averaging: equal weight for each class
  - Micro-averaging: equal weight for each instance
- See worked-out example on next slide

Multiclass classification

		guess			
		Y	N	U	
gold	Y	67	4	31	102
	N	1	16	4	21
	U	7	7	46	60
		75	27	81	183

class	precision		
Y	67/75	=	89.3%
N	16/27	=	59.3%
U	46/81	=	56.8%

Macro-averaged precision:

$$\frac{89.3 + 59.3 + 56.8}{3} = 68.5\%$$

Micro-averaged precision:

$$\frac{75 \cdot 89.3 + 27 \cdot 59.3 + 81 \cdot 56.8}{183} = 70.5\%$$

Evaluation metrics for retrieval

- Retrieval & recommendation problems
  - Very large space of possible outputs, many good answers
  - But outputs are simple (URLs, object ids), not structured
- Can be formulated as binary classification (of *relevance*)
- Problem: can't identify all positive items in advance
  - So, can't assess recall — look at *coverage* instead
  - Even precision is tricky, may require semi-manual process
- Evaluation metrics for ranked retrieval
  - Precision@k
  - Mean average precision (MAP)
  - Discounted cumulative gain

Evaluation metrics for complex outputs

- If outputs are numerous *and* complex, evaluation is trickier
  - Text (e.g., automatic summaries)
  - Tree structures (e.g., syntactic or semantic parses)
  - Grid structure (e.g., alignments)
- System outputs are unlikely to match gold standard exactly
- One option: manual eval — but slow, costly, subjective
- Another option: *approximate* comparison to gold standard
  - Give partial credit for partial matches
  - Text: n-gram overlap (ROUGE)
  - Tree structures: precision & recall over subtrees
  - Grid structures: precision & recall over pairs

Evaluation metrics for clustering

- Pairwise metrics (Hatzivassiloglou & McKeown 1993)
  - Reformulate as binary classification over *pairs* of items
  - Compute & report precision, recall, F1, MCC, ... as desired
- B^3 metrics (Bagga & Baldwin 1998)
  - Reformulate as a *set* of binary classification tasks, one per item
  - For each item, predict whether other items are in same cluster
  - Average per-item results over items (micro) or clusters (macro)
- Intrusion tasks
  - In predicted clusters, replace one item with random “intruder”
  - Measure human raters’ ability to identify intruder
- See Homework 7, Yao et al. 2012

Other evaluation metrics

- Regression problems
  - When the output is a real number
  - Pearson's R
  - Mean squared error

- Ranking problems
  - When the output is a rank
  - Spearman's rho
  - Kendall's tau
  - Mean reciprocal rank

Agenda

1. Introduction
2. Kinds of evaluation
3. Data management
4. Evaluation metrics
5. Comparative evaluations
6. Other aspects of evaluation
7. Conclusion

Comparative evaluation

- Say your model scores 77% on your chosen evaluation metric
- *Is that good? Is it bad?*
- You (& your readers) can't know unless you make comparisons
  - Baselines
  - Upper bounds
  - Previous work
  - Different variants of your model
- Comparisons are the *rows* of your main results table
  - Evaluation metrics are the columns
- Comparisons demand statistical significance testing!

Baselines

- 77% doesn't look so good if a blindfolded mule can get 73%
- Results without baseline comparisons are meaningless
- Weak baselines: performance of zero-knowledge systems
  - Systems which use no information about the specific instance
  - Example: random guessing models
  - Example: most-frequent class (MFC) models
- Strong baselines: performance of easily-implemented systems
  - Systems which can be implemented in an hour or less
  - WSD example: Lesk algorithm
  - RTE example: bag-of-words

Baselines example

word	#s	#ex	baselines: MFS	baselines: LeskC	word sense disambig.
argument	2	114	70.17%	73.63%	**89.47%**
arm	3	291	61.85%	69.31%	**84.87%**
atmosphere	3	773	54.33%	56.62%	**71.66%**
bank	3	1074	**97.20%**	**97.20%**	**97.20%**
bar	10	1108	47.38%	68.09%	**83.12%**
chair	3	194	67.57%	65.78%	**80.92%**
channel	5	366	51.09%	52.50%	**71.85%**
circuit	4	327	85.32%	85.62%	**87.15%**
degree	7	849	58.77%	73.05%	**85.98%**
difference	2	24	**75.00%**	**75.00%**	**75.00%**
disc	3	73	52.05%	52.05%	**71.23%**

from Mihalcea 2007

Upper bounds

- 77% doesn't look so bad if a even human expert gets only 83%
- *Plausible, defensible* upper bounds can flatter your results
- Human performance is often taken as an upper bound
  - Or inter-annotator agreement (for subjective labels)
  - (BTW, if you annotate your own data, report the kappa statistic)
  - If humans agree on only 83%, how can machines ever do better?
  - But in some tasks, machines outperform humans! (Ott et al. 2011)
- Also useful: oracle experiments
  - Supply gold output for some component of pipeline (e.g., parser)
  - Let algorithm access some information it wouldn't usually have
  - Can illuminate the system's operation, strengths & weaknesses

Comparisons to previous work

- Desirable, but not always possible — you may be a pioneer!
- Easy: same problem, same test data, same evaluation metric
  - Just copy results from previous work into your results table
  - The norm in tasks with standard data sets: ACE, Geo880, RTE, ...
- Harder: same problem, but different data, or different metric
  - Maybe you can obtain their code, and evaluate in your setup?
  - Maybe you can reimplement their system? Or an approximation?
- Hardest: new problem, new data set
  - Example: double entendre identification (Kiddon & Brun 2011)
  - Make your data set publicly available!
  - Let future researchers can compare to *you*

Different variants of your model

- Helps to shed light your model's strengths & weaknesses
- Lots of elements can be varied
  - Quantity, corpus, or genre of training data
  - Active feature categories
  - Classifier type or clustering algorithm
  - VSMs: distance metric, normalization method, ...
  - Smoothing / regularization parameters

Relative improvements

- It may be preferable to express improvements in *relative* terms
  - Say baseline was 60%, and your model achieved 75%
  - Absolute gain: 15%
  - Relative improvement: 25%
  - Relative error reduction: 37.5%

- Can be more informative (as well as more flattering!)
  - Previous work: 92.1%
  - Your model: 92.9%
  - Absolute gain: 0.8%
  - Relative error reduction: 10.1%

Statistical significance testing

- Pet peeve: small gains reported as fact w/o significance testing
  - "… outperforms previous approaches …"
  - "… demonstrates that word features help …"
- How likely is the gain you observed, under the null hypothesis?
  - Namely: model is no better than baseline, and gain is due to chance
- Crude solution: estimate variance using 10CV, or "the bootstrap"
- Analytic methods: McNemar's paired test, many others ...
- Monte Carlo methods: approximate randomization
  - Easy to implement, reliable, principled
  - Highly recommended reading: http://masanjin.net/sigtest.pdf

Agenda

1. Introduction
2. Kinds of evaluation
3. Data management
4. Evaluation metrics
5. Comparative evaluations
6. Other aspects of evaluation
7. Conclusion

Learning curve example

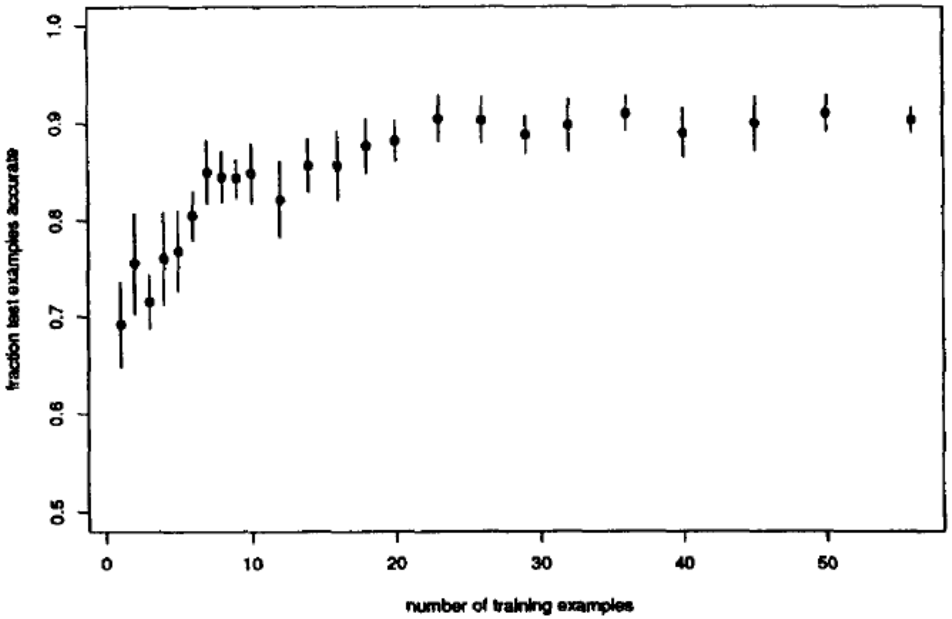
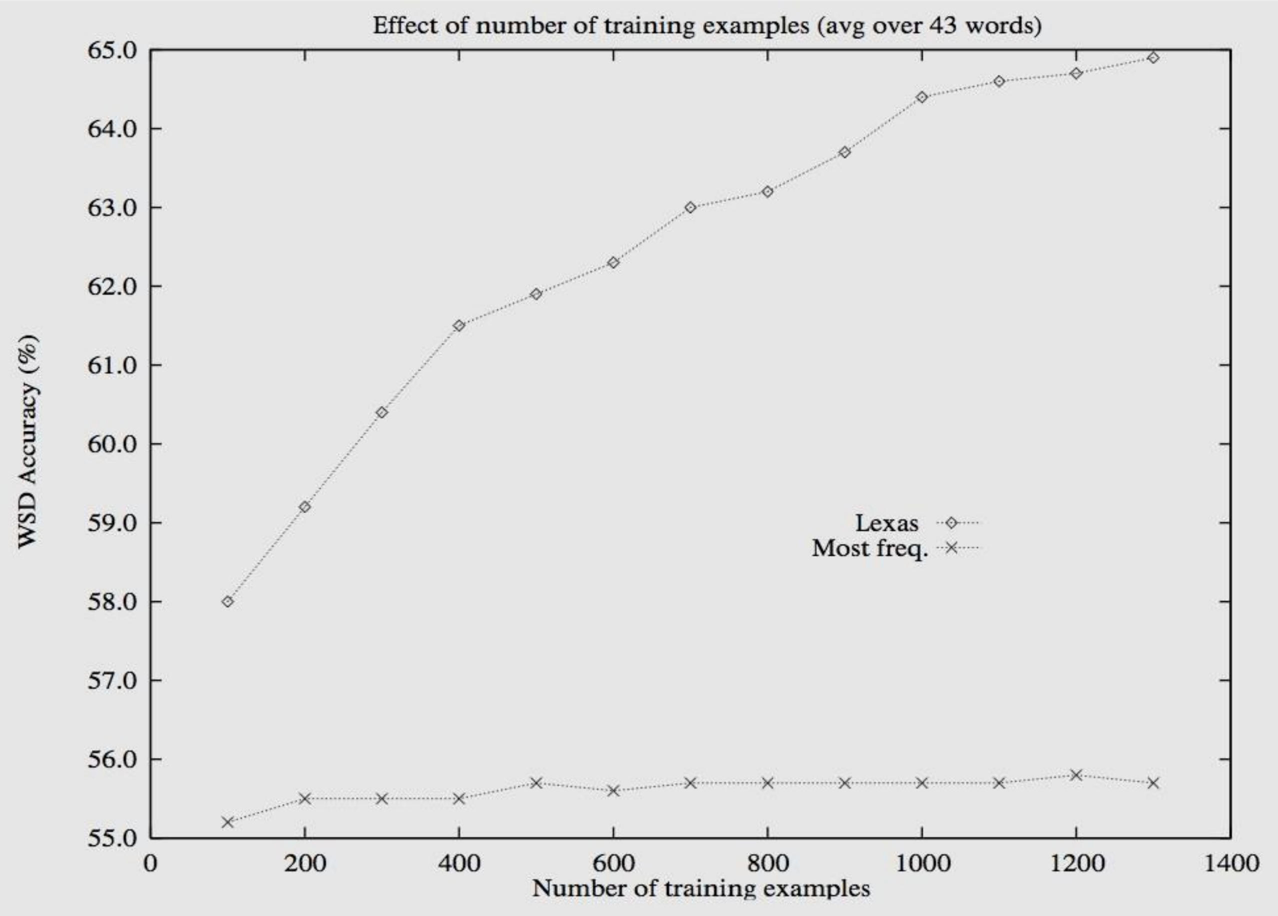


Figure IV. Just a few training examples do surprisingly well.
The horizontal axis shows the number of examples used in training while the vertical scale shows the mean percent correct in six disambiguations. The performance increases rapidly for the first few examples, and seems to have reached a maximum by 50 or 60 examples.

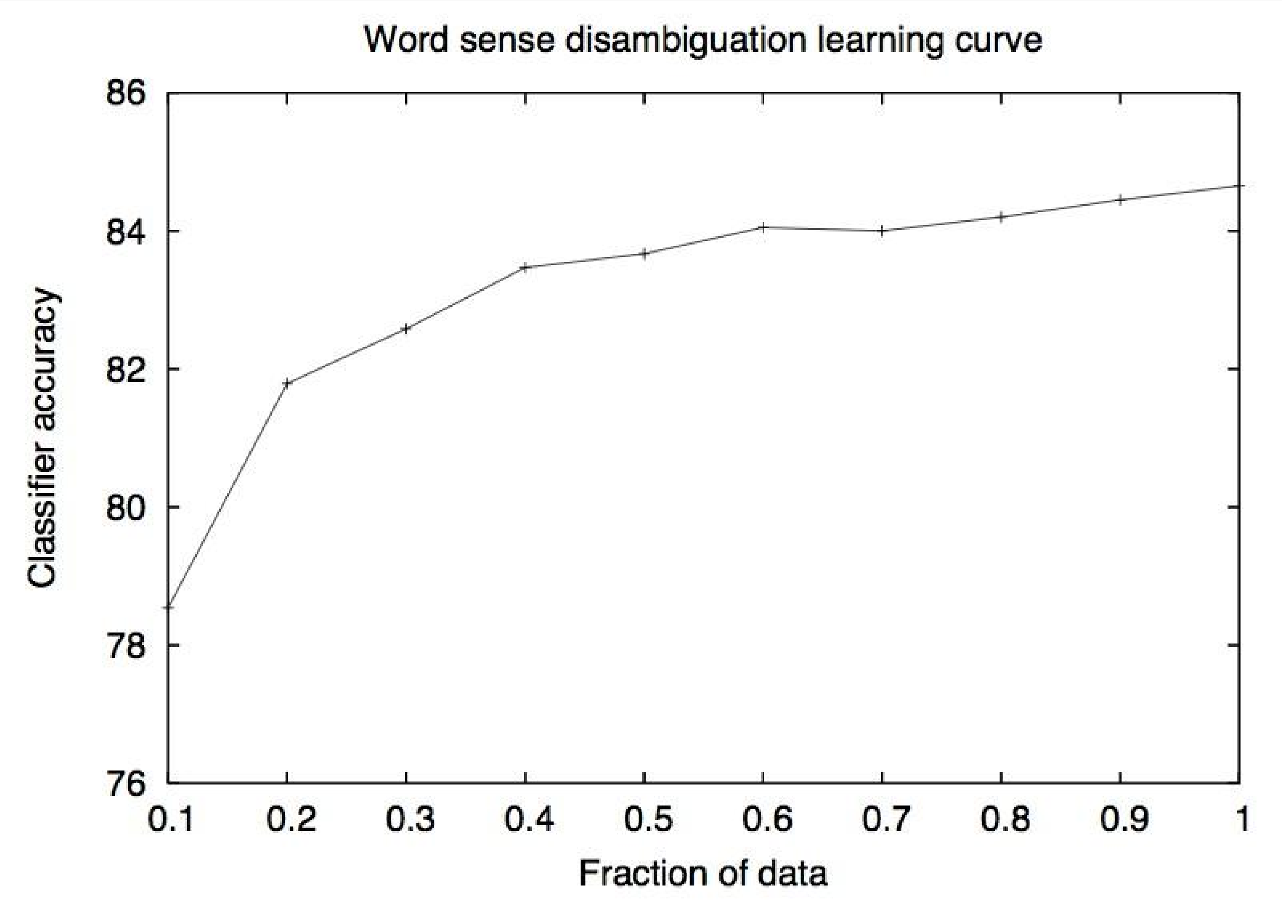
from Gale, Church, & Yarowsky 1992

Learning curve example



from Ng & Zelle 1997

Learning curve example



from [Mihalcea 2007]

Learning curves

- Plot evaluation metric as function of amount of training data
- May include multiple variants of model (e.g. classifier types)
- Provides insight into learning properties of model
- Pop quiz: what does it mean if ...
  - ... the curve is flat and never climbs?
  - ... the curve climbs and doesn't ever level off?
  - ... the curve climbs at first, but levels off quite soon?

Feature analysis

- Goal: understand which features are most informative
- Easy, but potentially misleading: list high-weight features
    - Implicitly assumes that features are independent
- Per-feature statistical measures
    - E.g., chi-square, information gain
    - Again, ignores potential feature interactions
- Ablation (or addition) tests
    - Progressively knock out (or add) (categories of) features
    - Do comparative evaluations at each step — often expensive!
- L1 regularization, Lasso, & other feature selection algorithms
    - Which features are selected? What are the regularization paths?

Example: high-weight features

Relation	Feature type	Left window	NE1	Middle	NE2	Right window
/architecture/structure/architect	LEX$\curvearrowleft$		ORG	, the designer of the	PER	
	SYN	designed $\Uparrow_s$	ORG	$\Uparrow_s$ designed $\Downarrow_{by-subj}$ by $\Downarrow_{pcn}$	PER	$\Uparrow_s$ designed
/book/author/works_written	LEX		PER	s novel	ORG	
	SYN		PER	$\Uparrow_{pcn}$ by $\Uparrow_{mod}$ story $\Uparrow_{pred}$ is $\Downarrow_s$	ORG	
/book/book_edition/author_editor	LEX$\curvearrowleft$		ORG	s novel	PER	
	SYN		PER	$\Uparrow_{nn}$ series $\Downarrow_{gen}$	PER	
/business/company/founders	LEX		ORG	co - founder	PER	
	SYN		ORG	$\Uparrow_{nn}$ owner $\Downarrow_{person}$	PER	
/business/company/place_founded	LEX$\curvearrowleft$		ORG	- based	LOC	
	SYN		ORG	$\Uparrow_s$ founded $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	
/film/film/country	LEX		PER	, released in	LOC	
	SYN	opened $\Uparrow_s$	ORG	$\Uparrow_s$ opened $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ opened
/geography/river/mouth	LEX		LOC	, which flows into the	LOC	
	SYN	the $\Downarrow_{det}$	LOC	$\Uparrow_s$ is $\Downarrow_{pred}$ tributary $\Downarrow_{mod}$ of $\Downarrow_{pcn}$	LOC	$\Downarrow_{det}$ the
/government/political_party/country	LEX$\curvearrowleft$		ORG	politician of the	LOC	
	SYN	candidate $\Uparrow_{nn}$	ORG	$\Uparrow_{nn}$ candidate $\Downarrow_{mod}$ for $\Downarrow_{pcn}$	LOC	$\Uparrow_{nn}$ candidate
/influence/influence_node/influenced	LEX$\curvearrowleft$		PER	, a student of	PER	
	SYN	of $\Uparrow_{pcn}$	PER	$\Uparrow_{pcn}$ of $\Uparrow_{mod}$ student $\Uparrow_{appo}$	PER	$\Uparrow_{pcn}$ of
/language/human_language/region	LEX		LOC	- speaking areas of	LOC	
	SYN		LOC	$\Uparrow_{lex-mod}$ speaking areas $\Downarrow_{mod}$ of $\Downarrow_{pcn}$	LOC	
/music/artist/origin	LEX$\curvearrowleft$		ORG	based band	LOC	
	SYN	is $\Uparrow_s$	ORG	$\Uparrow_s$ is $\Downarrow_{pred}$ band $\Downarrow_{mod}$ from $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ is
/people/deceased_person/place_of_death	LEX		PER	died in	LOC	
	SYN	hanged $\Uparrow_s$	PER	$\Uparrow_s$ hanged $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	$\Uparrow_s$ hanged
/people/person/nationality	LEX		PER	is a citizen of	LOC	
	SYN		PER	$\Downarrow_{mod}$ from $\Downarrow_{pcn}$	LOC	
/people/person/parents	LEX		PER	, son of	PER	
	SYN	father $\Uparrow_{gen}$	PER	$\Uparrow_{gen}$ father $\Downarrow_{person}$	PER	$\Uparrow_{gen}$ father
/people/person/place_of_birth	LEX$\curvearrowleft$		PER	is the birthplace of	PER	
	SYN		PER	$\Uparrow_s$ born $\Downarrow_{mod}$ in $\Downarrow_{pcn}$	LOC	
/people/person/religion	LEX		PER	embraced	LOC	
	SYN	convert $\Downarrow_{appo}$	PER	$\Downarrow_{appo}$ convert $\Downarrow_{mod}$ to $\Downarrow_{pcn}$	LOC	$\Downarrow_{appo}$ convert

Table 4: Examples of high-weight features for several relations. Key: SYN = syntactic feature; LEX = lexical feature; $\curvearrowleft$ = reversed; NE# = named entity tag of entity.

from Mintz et al. 2009

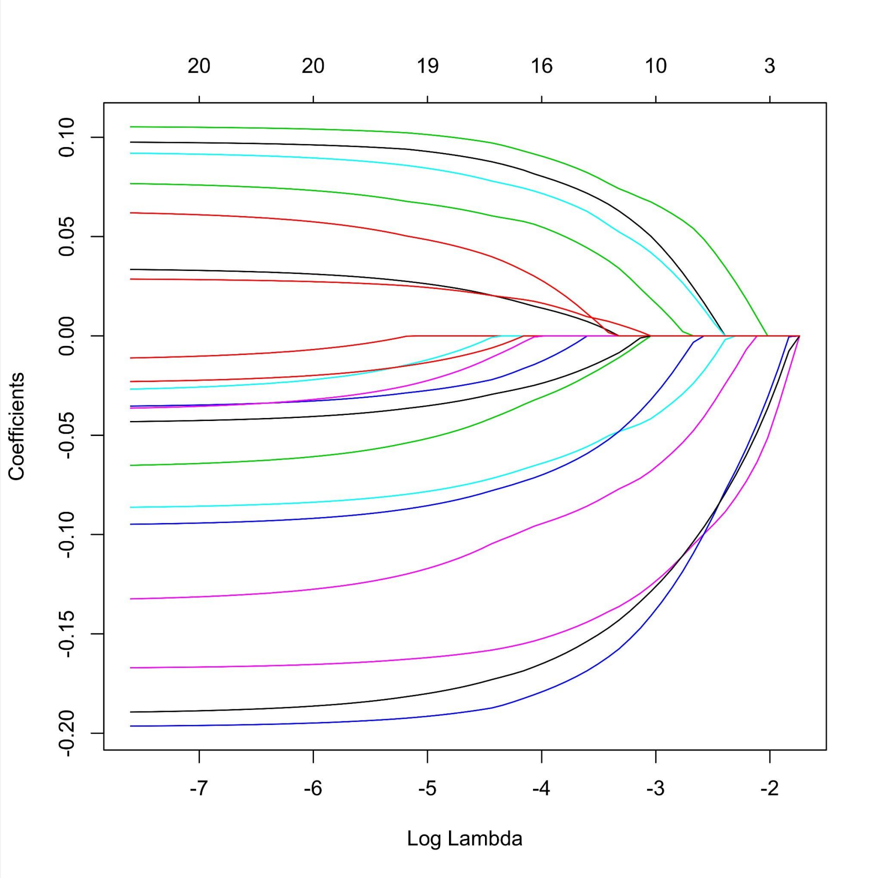
Example: feature addition tests

Features	P	R	F
Words	69.2	23.7	35.3
+Entity Type	67.1	32.1	43.4
+Mention Level	67.1	33.0	44.2
+Overlap	57.4	40.9	47.8
+Chunking	61.5	46.5	53.0
+Dependency Tree	62.1	47.2	53.6
+Parse Tree	62.3	47.6	54.0
+Semantic Resources	63.1	49.5	55.5

Table 2: Contribution of different features over 43 relation subtypes in the test data

from Zhou et al. 2005

Example: regularization paths



made using the glmnet package in R

Error analysis

- Analyze and categorize specific errors (on *dev* data, not test!)
- A form of *qualitative* evaluation — yet indispensable!
- During development (formative evaluation):
  - Examine individual mistakes, group into categories
  - Can be helpful to focus on FPs, FNs, common confusions
  - Brainstorm remedies for common categories of error
  - A key driver of iterative cycles of feature engineering
- In your report (summative evaluation):
  - Describe common categories of errors, exhibit specific examples
  - Aid the reader in understanding limitations of your approach
  - Highlight opportunities for future work

Error analysis example

4.3 Error Analysis

We also closely analyze the pairwise errors that we encounter when comparing against Freebase labels. Some errors arise because one instance can have multiple labels, as we explained in Section 4.1. One example is the following: Our approach predicts that (*News Corporation*, buy, *MySpace*) and (*Dow Jones & Company*, the parent of, *The Wall Street Journal*) are in one relation. In Freebase, one is labeled as "/organization/parent/child", the other is labeled as "/book/newspaper_owner/newspapers_owned". The latter is a sub-relation of the former. We can overcome this issue by introducing hierarchies in relation labels.

Some errors are caused by selecting the incorrect sense for an entity pair of a path. For instance, we put (*Kenny Smith*, who grew up in, *Queens*) and (*Phil Jackson*, return to, *Los Angeles Lakers*) into

from Yao et al. 2012

Agenda

1. Introduction
2. Kinds of evaluation
3. Data management
4. Evaluation metrics
5. Comparative evaluations
6. Other aspects of evaluation
7. Conclusion

Don’t fear negative results

Research is the process of going up alleys to see if they are blind.
— Marston Bates, American zoologist, 1906-1974

- Sometimes the results aren’t as good as you’d like
  - Sometimes you can’t show a statistically significant gain
  - Sometimes you can’t even beat the weak baseline :-(
- Your research work can still have value!
  - Especially if what you tried was a reasonable thing to try
  - Save future researchers from going up the same blind alleys
- Resist the temptation to optimize on test data
  - This is basically intellectual fraud

Plan for evaluation *early*

Evaluation should not be merely an afterthought;
it must be an integral part of designing a research project.

You can't aim if you don't have a target;
you can't optimize if you don't have an objective function.

First decide how to measure success;
then pursue it relentlessly!

Whoa, dude, that's some serious Yoda sh