Project planning & system evaluation

Bill MacCartney

CS224U

30 April 2018

Research workshop sessions

Today	Project planning & system eval
May 2	Active learning, crowdsourcing, methods for data annotation
May 23	Experimental methods
May 30	Writing up & presenting your work

Final project timeline

May 7	Due: Lit review (15%)
May 28	Due: Project milestone (10%)
June 6	Due: Oral presentations (5%)
June 13	Due: Final project paper (30%)

Forming teams

- You can work in teams of size 1, 2, or 3, but ...
- We heartily encourage teams of 3!
- Collaboration is the norm in scientific research, and in engineering and product development
- You can just get a lot more done: build bigger systems, explore more alternative choices, ...
- Leverage Piazza to find teammates

Goals for today

- Get you thinking concretely about what you want to accomplish
- Identify productive steps you can take even if you're still deciding on a topic or approach
- Try to help you avoid common pitfalls for projects
- Emphasize the importance of planning for system evaluation *early*
- Start building momentum for the final project!!

Inspiration

It’s nice if you do a great job and earn an A on your final project, but let’s think bigger:

- Many important and influential ideas, insights, and algorithms began as class projects
- Getting the best research-oriented jobs will likely involve giving a job talk. Your project can be the basis for one
- You can help out the scientific community by supplying data, code, and results (including things that didn’t work!)

Inspiring past projects

See https://cs224u.stanford.edu/restricted/past-final-projects/

- Semantic role labeling
- Unsupervised relation extraction
- Solving standardized test problems
- Humor detection
- Biomedical NER
- Sentiment analysis in political contexts
- Learning narrative schemas
- Supervised and unsupervised compositional semantics
- …

Agenda

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- Conclusion

The literature review

- A short (~6-page) single-spaced paper summarizing and synthesizing several papers in the area of your final project.
- Groups of one should review 5 papers; groups of two, 7 papers; and groups of three, 9 papers.
- Preferably fuel for the final project, but graded on its own terms.

The lit review: what to include

Tips on major things to include:

- General problem / task definition
- Concise summaries of the papers
- Compare & contrast approaches (most important!)
- Future work: what remains undone?

More details on the course website:
http://web.stanford.edu/class/cs224u/projects.html

See the end of this slideshow for the commenting/evaluating rubric that the teaching team will be using for lit reviews.

Our hopes

- The lit review research suggests baselines and approaches.
- The lit review helps us understand your project goals.
- We'll be able to suggest additional things to read.
- The prose itself can be modified for inclusion in your final paper.

Finding the literature

The relevant fields are extremely well-organized when it comes to collecting their papers and making them accessible:

- ACL Anthology: http://www.aclweb.org/anthology/
- ACL Anthology Searchbench: http://aclasb.dfki.de/
- ACM Digital Library: http://dl.acm.org/
- arXiv: http://arxiv.org/
- Google Scholar: http://scholar.google.com/

Best-first search algorithm

Until you get a core set of lit review papers:

1. Do a keyword search on the ACL Anthology
2. Download the papers that seem most relevant
3. Skim the abstracts, intros, & previous work sections
4. Identify papers that look relevant, appear often, & have lots of citations on Google Scholar
5. Download those papers
6. Return to step 3

Agenda

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- Conclusion

The importance of data

- Your investigation should be *empirical* — i.e., data-driven
- We are scientists!
  - Well, or engineers — either way, we're empiricists!
  - Not poets or philosophers :-)
- 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

Sources of data

Three strategies for obtaining data:

1. Find it (the easiest way!)
2. Create it (the laborious way)
3. Pay others to create it (the expensive way)

(Our discussion will focus primarily on labeled data for supervised learning, but applies to unlabeled data too.)

Finding datasets

Linguistic Data Consortium: http://www.ldc.upenn.edu/

- Very large and diverse archive
- Especially rich in annotated data
- Expensive (but often free for Stanford)

Other useful starting points:

- http://kevinchai.net/datasets
- https://datahub.io/dataset?tags=nlp
- http://research.microsoft.com/en-US/projects/data-science-initiative/datasets.aspx

Stanford Linguistics corpora

- We subscribe to the LDC and so have most of their datasets: https://linguistics.stanford.edu/resources/corpora/corpus-inventory
- To get access, follow the instructions at this page: https://linguistics.stanford.edu/resources/corpora/accessing-corpora
- When you write to the corpus TA, cc the CS224U course staff address. Don't forget this step!
- Write from your Stanford address. That will help the corpus TA figure out who you are and how to grant you access.

Some NLU datasets (open web)

- Stanford Question Answering Dataset (SQuAD): https://rajpurkar.github.io/SQuAD-explorer/
- Google Wikilinks corpus: http://www.iesl.cs.umass.edu/data/data-wiki-links
- Google Ngrams data: https://books.google.com/ngrams
- PPDB: The Paraphrase Database: http://paraphrase.org/#/download
- WikiAnswers Paraphrase Corpus: http://knowitall.cs.washington.edu/paralex/
- The SemEval-2014 SICK dataset: http://alt.qcri.org/semeval2014/task1/
- The Stanford NLI Corpus: http://nlp.stanford.edu/projects/snli/
- Abstract Meaning Representation (AMR) corpora: http://amr.isi.edu/download.html
- Winograd schemas: http://www.cs.nyu.edu/faculty/davise/papers/WS.html
- WebQuestions (semantic parsing): http://nlp.stanford.edu/software/sempre/

More NLU datasets (open web)

- Wikipedia data dumps: http://en.wikipedia.org/wiki/Wikipedia:Database_download
- Stack Exchange data dumps: https://archive.org/details/stackexchange
- Switchboard Dialog Act Corpus: http://www.stanford.edu/~jurafsky/ws97/
- Pranav Anand & co.: http://people.ucsc.edu/~panand/data.php
  - Internet Argument Corpus
  - Annotated political TV ads
  - Focus of negation corpus
  - Persuasion corpus (blogs)
- Datasets that Chris has made available as part of other courses and projects:
  - Extracting social meaning and sentiment: http://nasslli2012.christopherpotts.net
  - Computational pragmatics: http://compprag.christopherpotts.net
  - The Cards dialogue corpus: http://cardscorpus.christopherpotts.net

Some NLU datasets (on AFS)

Get access from the corpus TA, as described earlier:

- Nate Chambers' de-duped and dependency parsed NYT section of Gigaword: /afs/ir/data/linguistic-data/GigawordNYT
- Some data sets from Chris:
  - /afs/ir/data/linguistic-data/mnt/mnt4/PottsCorpora README.txt, Twitter.tgz, imdb-english-combined.tgz, opentable-english-processed.zip
  - /afs/ir/data/linguistic-data/mnt/mnt9/PottsCorpora opposingviews, product-reviews, weblogs
- Twitter data collected and organized by Moritz (former CS224Uer!) /afs/ir/data/linguistic-data/mnt/mnt3/TwitterTopics/

Scraping

- Link structure is often regular (reflecting database structure)
- If you figure out the structure, you can often get lots of data!
- Once you have local copies of the pages:
  - Beautiful Soup (Python) is a powerful tool for parsing DOMs
  - Readability offers an API for extracting text from webpages
- Use rate limiting / request throttling !!!!!
- Read site policies! Be a good citizen! Don't get yourself (or your school) banned! Don't go to jail! You will not like it.
- For more on crawler etiquette, see Manning et al. 2009 (http://nlp.stanford.edu/IR-book/)

Agenda

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- Conclusion

Project set-up

Now that you've got your dataset more or less finalized, you can start building stuff and doing experiments!

Automatic annotation tools

- If you need additional structure — POS tags, named-entity tags, parses, etc. — add it now.
- The Stanford NLP group has released lots of software for doing this: http://nlp.stanford.edu/software/index.shtml
- Can be used as libraries in Java/Scala, or from the command line.
- Check out CoreNLP in particular — amazing! https://stanfordnlp.github.io/CoreNLP/
- Many other alternatives, including NLTK and spaCy in Python

Off-the-shelf modeling tools

While there's some value in implementing algorithms yourself, it's labor intensive and could seriously delay your project. We advise using existing tools whenever possible:

- fastText (C++): https://fasttext.cc/
- spaCy (Python): https://spacy.io/
- NLTK (Python): http://nltk.org/
- scikit-learn (Python): http://scikit-learn.org/
- Gensim (Python): http://radimrehurek.com/gensim/
- Stanford Classifier (Java): http://nlp.stanford.edu/software/classifier.shtml
- MALLET (Java): http://mallet.cs.umass.edu/
- FACTORIE (Scala): http://factorie.cs.umass.edu/
- LingPipe (Java): http://alias-i.com/lingpipe/
- GATE (Java): http://gate.ac.uk/
- Lucene (Java): http://lucene.apache.org/core/

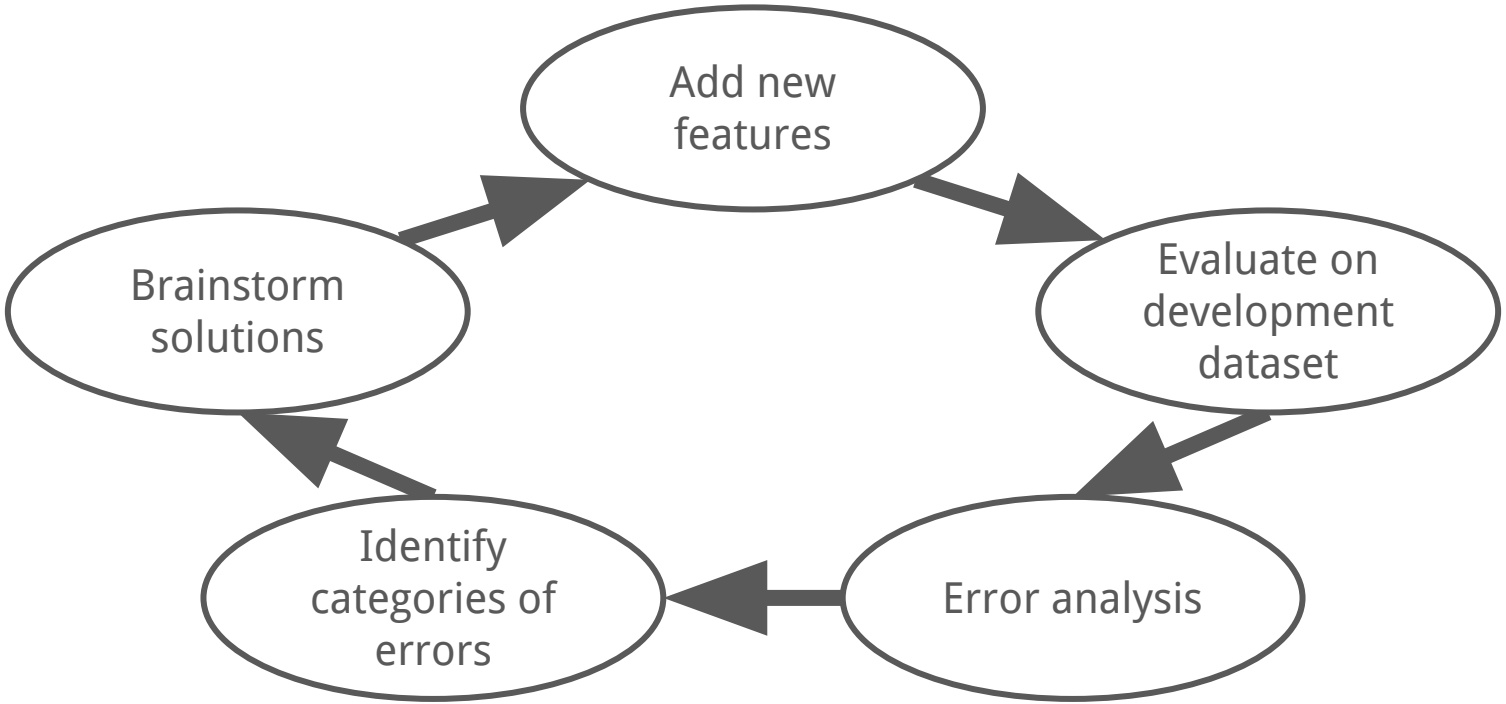
Iterative development

Launch & iterate!

- Get a baseline system running on real data ASAP
- Implement an evaluation
  - Ideally, an automatic, quantitative evaluation
  - But could be more informal if necessary
- Hill-climb on your objective function
- Add features, tune parameters, fiddle with model architecture, …
- As you go, document what you've learned so far

Goal: research as an anytime algorithm: have some result to show at every stage

The feature engineering cycle



More development tips

- Construct a tiny toy dataset for development
  - Facilitates understanding model behavior, finding bugs
- Consider ensemble methods
  - Develop multiple models with complementary expertise
  - Combine via max/min/mean/sum, voting, meta-classifier, …
- Grid search in parameter space can be useful
  - Esp. for “hyperparameters”
  - Esp. when parameters are few and evaluation is fast
  - A kind of informal machine learning

Agenda

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- Conclusion

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!

See also our notebook on evaluation in NLP

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?

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 — tables & graphs & charts, oh my!
- But qualitative evaluations are useful too!
  - Examples of system outputs
  - Error analysis
  - Visualizations
  - Interactive demos
    - A great way to gain visibility and impact for your work
    - Examples: OpenIE (relation extraction), Deeply Moving (sentiment)
- A tremendous aid to your readers' understanding!

Example of system outputs

Relation name	New instance
/location/location/contains	Paris, Montmartre
/location/location/contains	Ontario, Fort Erie
/music/artist/origin	Mighty Wagon, Cincinnati
/people/deceased_person/place_of_death	Fyodor Kamensky, Clearwater
/people/person/nationality	Marianne Yvonne Heemskerk, Netherlands
/people/person/place_of_birth	Wavell Wayne Hinds, Kingston
/book/author/works_written	Upton Sinclair, Lanny Budd
/business/company/founders	WWE, Vince McMahon
/people/person/profession	Thomas Mellon, judge

Table 1: Ten relation instances extracted by our system that did not appear in Freebase.

from Mintz et al. 2009

Example of visualization

from Joseph Turian

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)

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 which uses your model
  - 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

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- Conclusion

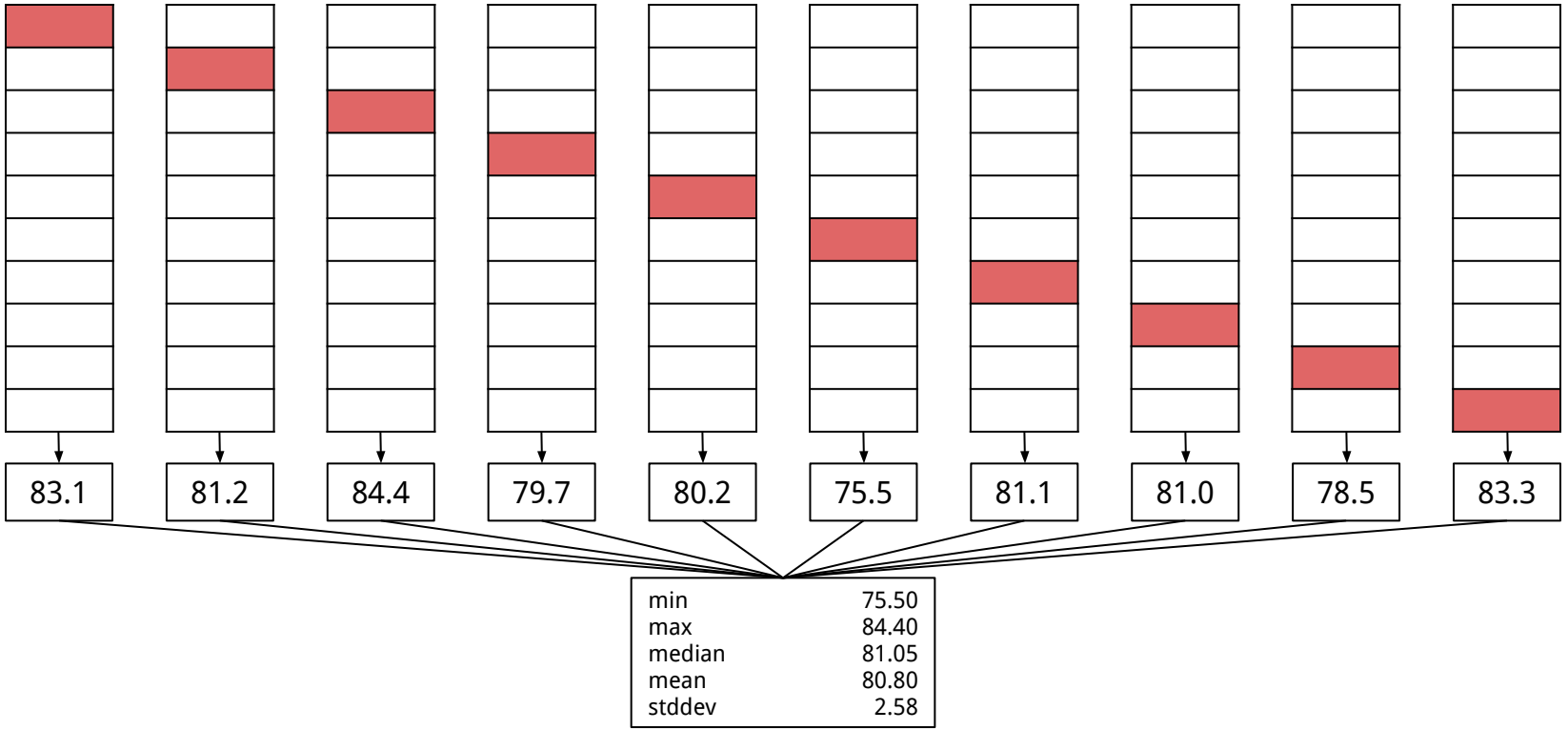
The train/test split

- 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
  - If you're using a standard dataset, the split is often predefined
  - Don't evaluate on it until the very end! 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

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- Conclusion

Evaluation metrics

- An evaluation metric is a function: model × data → $\mathbb{R}$
- 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
- See also our notebook on evaluation metrics in NLP

Evaluation metrics

Classification problems

- Contingency tables
- Confusion matrices
- Accuracy
- Precision & recall
- F-measure
- AUC (area under ROC curve)
- Sensitivity & specificity
- PPV & NPV (+/– predictive value)
- MCC (Matthews corr. coeff.)

Clustering problems

- Pairwise metrics
- B^3 metrics
- Intrusion tasks

Regression problems

- Pearson's R
- Mean squared error

Ranking problems

- Spearman's rho
- Kendall's tau
- Mean reciprocal rank

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

Agenda

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- 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
  - Sentiment example: hand-crafted lexicon with +1/–1 weights
  - NLI example: bag-of-words

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 datasets: SNLI, SQuAD, ...
- 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
  - Make your dataset publicly available!
  - Make your code 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% (yawn)
  - Relative error reduction: 10.1% (wow!)

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: https://cs.stanford.edu/~wmorgan/sigtest.pdf

Significant skepticism

Lately there's been some healthy skepticism about the value of p-values. For example:

http://www.nature.com/news/scientific-method-statistical-errors-1.14700

Lesson: $p < 0.05$ may not be a reliable indicator of a truly significant result.

But $p > 0.05$ still means you haven't proven s---.

And you should still do significance testing!

Still not significant

If the result ain't significant, just admit it!

No weasel words!

(barely) not statistically significant (p=0.052)
a borderline significant trend (p=0.09)
a certain trend toward significance (p=0.08)
a clear tendency to significance (p=0.052)
a clear, strong trend (p=0.09)
a decreasing trend (p=0.09)
a definite trend (p=0.08)
a distinct trend toward significance (p=0.07)
a favorable trend (p=0.09)
a favourable statistical trend (p=0.09)
a little significant (p<0.1)
a margin at the edge of significance (p=0.0608)
a marginal trend (p=0.09)
a marginal trend toward significance (p=0.052)
a marked trend (p=0.07)
a mild trend (p<0.09)
a near-significant trend (p=0.07)
a nonsignificant trend (p<0.1)
a notable trend (p<0.1)
a numerical increasing trend (p=0.09)
a numerical trend (p=0.09)
a positive trend (p=0.09)
a possible trend toward significance (p=0.052)
a pronounced trend (p=0.09)
a reliable trend (p=0.058)
a robust trend toward significance (p=0.0503)
a significant trend (p=0.09)

just lacked significance (p=0.053)
just marginally significant (p=0.0562)
just missing significance (p=0.07)
just on the verge of significance (p=0.06)
just outside levels of significance (p<0.08)
just outside the bounds of significance (p=0.06)
just outside the level of significance (p=0.0683)
just outside the limits of significance (p=0.06)
just short of significance (p=0.07)
just shy of significance (p=0.053)
just tendentially significant (p=0.056)
leaning towards significance (p=0.15)
leaning towards statistical significance (p=0.06)
likely to be significant (p=0.054)
loosely significant (p=0.10)
marginal significance (p=0.07)
marginally and negatively significant (p=0.08)
marginally insignificant (p=0.08)
marginally nonsignificant (p=0.096)
marginally outside the level of significance
marginally significant (p>=0.1)
marginally significant tendency (p=0.08)
marginally statistically significant (p=0.08)
may not be significant (p=0.06)
medium level of significance (p=0.051)
mildly significant (p=0.07)
moderately significant (p>0.11)

slightly significant (p=0.09)
somewhat marginally significant (p>0.055)
somewhat short of significance (p=0.07)
somewhat significant (p=0.23)
strong trend toward significance (p=0.08)
sufficiently close to significance (p=0.07)
suggestive of a significant trend (p=0.08)
suggestive of statistical significance (p=0.06)
suggestively significant (p=0.064)
tantalisingly close to significance (p=0.104)
technically not significant (p=0.06)
teetering on the brink of significance (p=0.06)
tended toward significance (p=0.13)
tentatively significant (p=0.107)
trend in a significant direction (p=0.09)
trending towards significant (p=0.099)
vaguely significant (p>0.2)
verging on significance (p=0.056)
very narrowly missed significance (p<0.06)
very nearly significant (p=0.0656)
very slightly non-significant (p=0.10)
very slightly significant (p<0.1)
virtually significant (p=0.059)
weak significance (p>0.10)
weakly significant (p=0.11)
weakly statistically significant (p=0.0557)
well-nigh significant (p=0.11)

http://mchankins.wordpress.com/2013/04/21/still-not-significant-2/

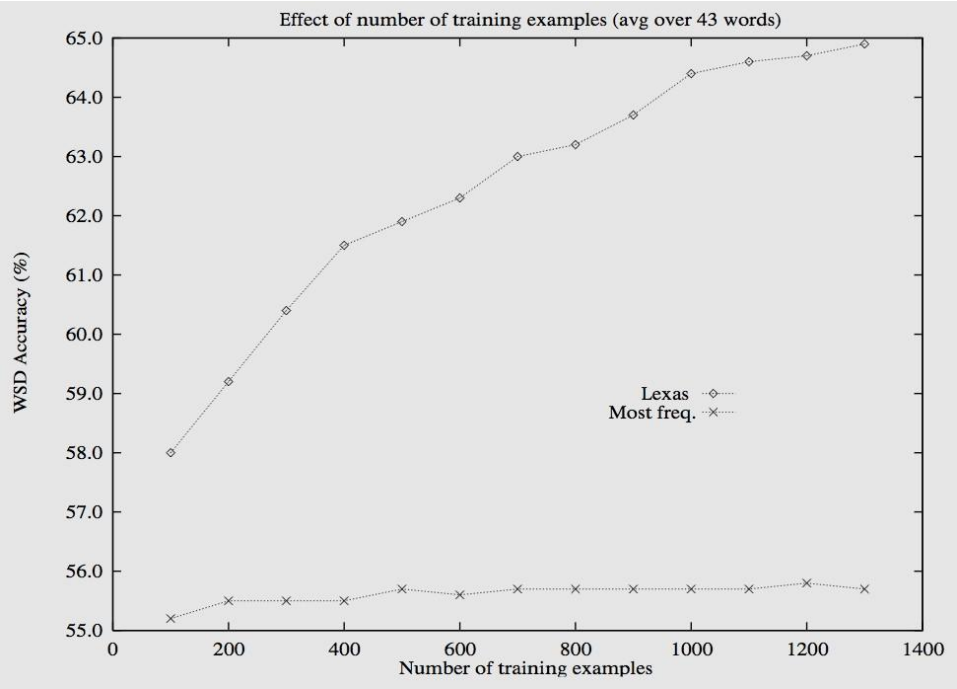
Agenda

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- Conclusion

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?

Learning curve example



from Ng & Zelle 1997

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

Visualizations

- Helpful in making multiple formal and informal comparisons, identify overlooked relationships
- t-SNE for 2d visualization of high-dimensional data: http://homepage.tudelft.nl/19j49/t-SNE.html
- Gephi: http://gephi.org/
- Visualization tools from Jeff Heer's group: https://homes.cs.washington.edu/~jheer/

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

Example: error analysis

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

- Overview
- Lit review
- Data sources
- Project set-up & development
- Evaluation
- Dataset management
- Evaluation metrics
- Comparative evaluations
- Other aspects of evaluation
- Conclusion

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

Game plan

- Form a team of 3 and choose a topic
- Survey previous work — lit review due May 7
- Identify data sources *soon*
- Leverage off-the-shelf tools where possible
- Launch & iterate — “anytime” research process
- Plan for evaluation early!

Lit review commenting/grading rubric

1. Is the general problem/task definition clearly articulated?
2. Do the summaries articulate the major contributions of each article and identify informative similarities and differences between the papers?
3. Are there existing, high-quality publicly available datasets for working in this area? If not, is data availability likely to be an obstacle to progress on the project?
4. Are the future work ideas feasible given the available data and time allotted? If yes, do any stand out as really good paths to follow? If not, can we formulate a workable path forward using the ideas in this lit review?
5. Other relevant work, or tips on where to look for relevant work?

Milestone commenting/grading rubric

1. Does the milestone clearly state the project's goal? Can you tell what hypotheses are being tested?
2. Is the milestone in the same area as the lit review? If not, is this a cause for concern, say, because it's not clear whether the team has built up the requisite expertise in this area?
3. Does the milestone give a representative picture of prior approaches and what they are like?
4. Does the milestone contain a clear progress report?
5. Has the team made sufficient progress that you feel confident they can complete an excellent project by June 13?

Final paper commenting/grading rubric

This is adapted from ACL-style reviewing forms. We don't give numerical ratings the way ACL reviewers do, but this does reflect similar values. Note: no discussion at all of the *strength* of the findings, but rather only of the *quality of the evaluation and reporting*.

1. Is the paper clear and well-written?
2. Is it clear what hypothesis or hypotheses are being tested, and do the experiments do a good job of testing it/them?
3. Are the evaluations solid, with good baselines, fair comparisons, and clear arguments for differences, or lack thereof, between systems?
4. Are the results clearly reported?
5. Are the authors making their code and/or data publicly available? (This is optional, but we want to support efforts at open science.)