Methods and metrics

Christopher Potts

Stanford Linguistics

CS224u: Natural language understanding

Overview

Goal: Help you with your projects

- Managing data
- Establishing baselines
- Comparing models
- Optimizing models
- Navigating tricky situations

Associated materials

- Evaluation metrics notebook: https://github.com/cgpotts/cs224u/blob/master/evaluation_metrics.ipynb
- scikit-learn guidance on model evaluation: http://scikit-learn.org/stable/modules/model_evaluation.html
- Evaluation methods notebook: https://github.com/cgpotts/cs224u/blob/master/evaluation_methods.ipynb
- Resnik and Lin 2010; Smith 2011, Appendix B

Your projects

1. We will never evaluate a project based on how "good" the results are.
    - Publication venues do this, because they have additional constraints on space that lead them to favor positive evidence for new developments over negative results.
    - In CS224u, we are not subject to this constraint, so we can do the right and good thing of valuing positive results, negative results, and everything in between.
2. We will evaluate your project on:
    - The appropriateness of the metrics.
    - The strength of the methods.
    - The extent to which the paper is open and clear-sighted about the limits of its findings.

Methods: How times have changed!

Circa 2010

1. Develop your complete system on tiny samples of your train data.
2. Once it is working, do regular cross-validation using only your train data
3. Evaluate only very occasionally on dev so that you don't hill climb on it.
4. In the final stages of your project, do a complete round of hyperparameter tuning using your dev data, select the best model, and evaluate it on test.

In 2023

1. Develop your complete system on tiny samples of your train data.
2. Either there is no train data or cross-validation would cost $20K and take six months.
3. Dev is frequently and crucially important to optimization, so 🤞 that it's a superb proxy for test.
4. Either hyperparameter tuning would cost $100K and take ten years or there are no hyperparameters but test runs cost $4K in API costs.

So what do we do?

1. The core tenets of the previous era remain perfectly sound.
2. But enforcing them has become impossible – only the richest organizations could follow them, and restricting participation in the field in that way would be terrible.
3. So: articulate your methods and the rationale behind them, including practical details.
4. Two rules should remain absolutely fixed:
   - Never do any model selection (even informally) based on test set evaluations.
   - Try to give all the systems you evaluate the best chance of success – never stack the deck in favor of a system you are advocating for.

Metrics: How times should change!

Strathern's Law: When a measure becomes a target, it ceases to be a good measure.

Leaderboards – the good

An objective basis for comparisons, creating opportunities for wild-seeming ideas to get a hearing.

Leaderboards – the bad

- Conflation of benchmark improvements with progress
- Conflation of benchmarks with empirical domains ("X is solved")
- Conflation of benchmark performance with capabilities

Metrics and application areas

- Missing a safety signal costs lives; human review is feasible
- Exemplars need to be found in a massive dataset
- Specific mistakes are deal-breakers; others hardly matter
- Cases need to be prioritized
- The solution needs to work over an aging cell network
- The solution cannot provide worse service to specific groups
- Specific predictions need to be blocked

Our (apparent) answer: F1 and friends

What we seem to value

The Values Encoded in Machine Learning Research

Abeba Birhane*
University College Dublin & Lero
Dublin, Ireland
abeba.birhane@ucdconnect.ie

Pratyusha Kalluri*
Stanford University
pkalluri@stanford.edu

Dallas Card*
Stanford University
dcard@stanford.edu

William Agnew*
University of Washington
wagnew3@cs.washington.edu

Ravit Dotan*
University of California, Berkeley
ravit.dotan@berkeley.edu

Michelle Bao*
Stanford University
baom@stanford.edu

What we seem to value

Selected 'Values encoded in ML research' from Birhane et al. (2021):

Performance

Efficiency

Interpretability (for researchers)

Applicability in the real world

Robustness

Scalability

Interpretability (for users)

Benificence

Privacy

Fairness

Justice

What we seem to value

Selected 'Values encoded in ML research' from Birhane et al. (2021):

Performance

Towards multidimensional leaderboards

Dynaboard: An Evaluation-As-A-Service Platform for Holistic Next-Generation Benchmarking

Led

DAWNBench: An End-to-End Deep Learning Benchmark and Competition

Cody Cole

EXPLAINABOARD: An Explainable Leaderboard for NLP

Pengfei Liu[1†], **Jinlan Fu**[2], **Yang Xiao**[2], **Weizhe Yuan**[1], **Shuaichen Chang**[3], **Junqi Dai**[2], **Yixin Liu**[1], **Zihuiwen Ye**[1], **Zi-Yi Dou**[1], **Graham Neubig**[1‡]
[1]Carnegie Mellon University, [2]Fudan University, [3]The Ohio State University,
†pliu3@cs.cmu.edu, ‡gneubig@cs.cmu.edu

Dodge et al. 2019; Ethayarajh and Jurafsky 2020

Dynascores

Model	Performance	Throughput	Memory	Fairness	Robustness	Dynascore
	8	2	2	2	2	
DeBERTa	76.25	4.47	6.97	88.33	90.06	45.92
ELECTRA-large	76.07	2.37	25.30	93.13	91.64	45.79
RoBERTa	69.67	6.88	6.17	88.32	86.10	42.54
ALBERT	68.63	6.85	2.54	87.44	80.90	41.74
BERT	57.14	6.70	5.55	91.45	80.81	36.07
BiDAF	53.48	10.71	3.60	80.79	77.03	33.96
Unrestricted T5	28.80	4.51	10.69	92.32	88.41	22.18
Return Context	5.99	89.80	1.10	95.97	91.61	15.47

Question answering

Ma et al. 2021; https://dynabench.org
https://github.com/cgpotts/cs224u/blob/main/dynascoring.ipynb

Dynascores

	8	1	1	5	1	
Model	Performance	Throughput	Memory	Fairness	Robustness	Dynascore
DeBERTa	76.25	4.47	6.97	88.33	90.06	46.70
ELECTRA-large	76.07	2.37	25.30	93.13	91.64	46.86
RoBERTa	69.67	6.88	6.17	88.32	86.10	43.37
ALBERT	68.63	6.85	2.54	87.44	80.90	42.66
BERT	57.14	6.70	5.55	91.45	80.81	37.17
BiDAF	53.48	10.71	3.60	80.79	77.03	34.62
Unrestricted T5	28.80	4.51	10.69	92.32	88.41	23.19
Return Context	5.99	89.80	1.10	95.97	91.61	14.29

Question answering

Ma et al. 2021; https://dynabench.org
https://github.com/cgpotts/cs224u/blob/main/dynascoring.ipynb

Turing Test results

A machine's behavior is intelligent if it can trick a human interrogator into thinking it is human using only conversation.

- Report from the first Turing Test (Shieber 1994): Shakespeare expert Cynthia Clay thrice misclassified as a computer.
- 2014 Turing Test event: AI Eugene Goostman ("13-year-old Ukrainian boy") passes!
- Google Duplex: An AI that routinely runs and wins Turing tests with service workers.
- Clark et al. (2021), "All That's 'Human' Is Not Gold"

Estimating human performance

Premise	Label	Hypothesis
A dog jumping	neutral	A dog wearing a sweater
turtle	contradiction	linguist
A photo of a race horse	?	A photo of an athlete
A chef using a barbecue	?	A person using a machine

Human response throughout: “Let’s discuss”

“Human performance” ≈ Average performance of harried crowdworkers doing a machine task repeatedly

Pavlick and Kwiatkowski 2019

Somewhere between accuracy and Turing tests

1. Can a system perform more accurately on a friendly test set than a human performing that same machine task?
   (Standard: scalable and familiar)
2. Can a system behave systematically (even if it's not accurate)?
3. Can a system assess its own confidence – know when not to make a prediction (Rajpurkar et al. 2018)?
4. Can a system make people happier and more productive?
5. Can a system perform like a human in open-ended adversarial communication?
   (Turing test: particular and thorny)

Times *are* changing!

Assessment today

- One-dimensional
- Largely insensitive to context (use-case)
- Terms set by the research community
- Opaque
- Tailored to machine tasks

Assessments in the future

- High-dimensional and fluid
- Highly sensitive to context (use-case)
- Terms set by the stakeholders
- Judgments ultimately made by users
- Tailored to human tasks

Classifier metrics

Overview

- Different evaluation metrics **encode different values**.
- Choosing a metric is a crucial aspect to experimental work.
- You should feel free to motivate new metrics and specific uses of existing metrics, depending on what your goals are.
- For established tasks, there is usually pressure to use specific metrics, but you should feel empowered to push back.
- Areas can stagnate due to poor metrics, so we must be vigilant!

Confusion matrices

		Predicted			
		pos	neg	neutral	Support
	pos	15	10	100	125
Gold	neg	10	15	10	35
	neutral	10	100	1000	1110

A threshold was imposed for these categorical predictions.

Accuracy

The correct predictions divided by the total number of examples.

		Predicted		
		pos	neg	neutral
Gold	pos	15	10	100
	neg	10	15	10
	neutral	10	100	1000

- Bounds: $[0, 1]$, with 0 the worst and 1 the best.
- Value encoded: how often is the system correct?
- Weaknesses:
  - No per-class metrics.
  - Failure to control for class size.

Accuracy and the cross-entropy loss

Cross-entropy loss

Accuracy is inversely proportional to the negative log-loss (a.k.a. cross entropy loss; sklearn link):

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{i,k}\log(p_{i,k})$$

KL-divergence

KL-divergence is an analogue of accuracy for soft labels:

$$D_{\mathrm{KL}}(y\|p)\sum_{k=1}^{K} y_k \log\left(\frac{y_k}{p_k}\right)$$

Where y is a "one-hot vector" with 1 at position k, this reduces to

$$\log\left(\frac{1}{p_k}\right) = -\log(p_k)$$

Precision

For class k: the correct predictions for k divided by the sum of all guesses for k.

		Predicted		
		pos	neg	neutral
	pos	**15**	10	100
Gold	neg	**10**	15	10
	neutral	**10**	100	1000
	Precision	0.43	0.12	0.90

Precision for pos: $15 \ / \ (15 + 10 + 10) = 0.43$

- Bounds: $[0, 1]$, with 0 the worst and 1 the best. (Caveat: undefined values resulting from dividing by 0 need to be mapped to 0.)
- Value encoded: penalize incorrect guesses.
- Weakness: Achieve high precision for k simply by rarely guessing k.

Recall

For class k: the correct predictions for k divided by the sum of all true members of k.

		Predicted			
		pos	neg	neutral	Recall
	pos	**15**	**10**	**100**	0.12
Gold	neg	10	15	10	0.43
	neutral	10	100	1000	0.90

Recall for pos: $15 \ / \ (15 + 10 + 100) = 0.12$

- Bounds: $[0, 1]$, with 0 the worst and 1 the best.
- Value encoded: penalize missed true cases.
- Weakness: Achieve high recall for k simply by always guessing k.

F scores

$$\mathrm{F}_{\beta}(k) = (\beta^2 + 1) \cdot \frac{\mathrm{Precision}(k) \cdot \mathrm{Recall}(k)}{(\beta^2 \cdot \mathrm{Precision}(k)) + \mathrm{Recall}(k)}$$

		Predicted			
		pos	neg	neutral	F_1
Gold	pos	15	10	100	0.19
	neg	10	15	10	0.19
	neutral	10	100	1000	0.90

- Bounds: $[0, 1]$, with 0 the worst and 1 the best; always between precision and recall.
- Value encoded: how much do predictions for k align with true instances of k, with β controlling the weight places on precision vs. recall
- Weaknesses:
  - No normalization for the size of the dataset.
  - Ignores the values off the row and column for k.

Averaging F scores

- Macro-averaging
- Weighted averaging
- Micro-averaging

Macro-averaged F scores

		Predicted			
		pos	neg	neutral	F_1
	pos	15	10	100	**0.19**
Gold	neg	10	15	10	**0.19**
	neutral	10	100	1000	**0.90**
					0.43

- Bounds: [0, 1], with 0 the worst and 1 the best.
- Value encoded: same values as F scores plus the assumption that all classes are equal.
- Weaknesses:
  - A classifier that does well only on small classes might not do well in the real world.
  - A classifier that does well only on large classes might do poorly on small but vital smaller ones.

Weighted average F scores

		Predicted				
		pos	neg	neutral	Support	F_1
Gold	pos	15	10	100	125	**0.19**
	neg	10	15	10	35	**0.19**
	neutral	10	100	1000	1110	**0.90**
						0.43

$$\frac{0.19 \cdot 125 + 0.19 \cdot 35 + 0.90 \cdot 1110}{125 + 35 + 1110}$$

- Bounds: $[0, 1]$, with 0 the worst and 1 the best.
- Value encoded: same values as F_β plus the assumption that class size matters.
- Weaknesses: Large classes will dominate.

Micro-averaged F scores

		Predicted pos	Predicted neg	Predicted neutral
Gold	pos	15	10	100
Gold	neg	10	15	10
Gold	neutral	10	100	1000

	yes	no
yes	15	110
no	20	1125

	yes	no
yes	15	20
no	110	1125

	yes	no
yes	1000	110
no	110	50

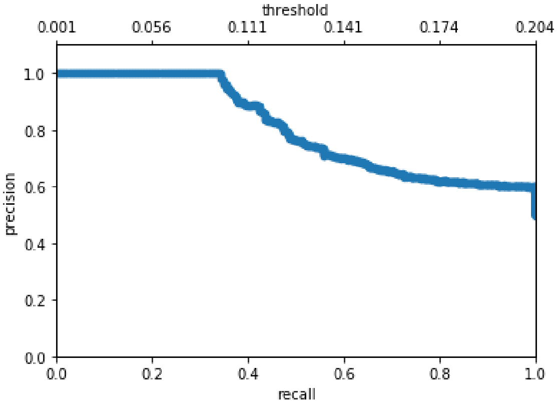
	yes	no	F_1
yes	1030	240	0.81
no	240	2300	0.91

Micro-averaged F scores

- Bounds: $[0, 1]$, with 0 the worst and 1 the best.
- Value encoded: Micro-averaged F_1 for "yes" $=$ accuracy.
- Weaknesses:
  - Same as for weighted F scores, plus
  - a score for "yes" and "no", hence no single summary number.

Precision–recall curves

Summarizes the relationship between precision and recall by using each predicted probability as a potential threshold:



Average precision provides a summary of the curve.

Generation metrics

Challenges

1. There is more than one effective way to say most things.
2. What are we measuring?
   - Fluency?
   - Truthfulness?
   - Communicative effectiveness?

Perplexity of a probability distribution

Perplexity

For a sequence $\mathbf{x} = [x_1, \ldots x_n]$ and probability distribution p:

$$\mathbf{PP}(p, \mathbf{x}) = \prod_{i=1}^{n} \left(\frac{1}{p(x_i)} \right)^{\frac{1}{n}}$$

Mean perplexity

For a corpus X of m examples:

$$\textbf{mean-PP}(p, X) = \exp\left(\frac{1}{m} \sum_{\mathbf{x} \in X} \log \mathbf{PP}(p, \mathbf{x}) \right)$$

Perplexity: Properties

- Bounds: $[1, \infty]$, with 1 best.
- Equivalent to the exponentiation of the cross-entropy loss.
- Value encoded: does the model assign high probability to the input sequence?
- Weaknesses:
  - Heavily dependent on the underlying vocabulary.
  - Doesn't allow comparisons between datasets.
  - Even comparisons between models are tricky.

Word-error rate: Definition

Edit distance

A measure of distance between strings. Word-error rate can be seen as a family of measures depending on the choice of distance measure.

Word-error rate

$$\mathbf{wer}(\mathbf{x}, \mathbf{pred}) = \frac{\text{distance}(\mathbf{x}, \mathbf{pred})}{\text{length}(\mathbf{x})}$$

Corpus word-error rate

For a corpus X:

$$\frac{\sum_{\mathbf{x} \in X} \text{distance}(\mathbf{x}, \mathbf{pred})}{\sum_{\mathbf{x} \in X} \text{length}(\mathbf{x})}$$

Word-error rate: Properties

- Bounds: $[0, \infty]$, with 0 the best.
- Value encoded: how aligned is the predicted sequence with the actual sequence – similar to F scores.
- Weaknesses:
  - Just one reference text.
  - A very syntactic notion – consider *It was good* vs. *It was not good.* vs. *It was great*

BLEU scores: Definition

Modified n-gram precision

Candidate:	`the the the the the the the`
Ref 1:	`the cat is on the mat`
Ref 2:	`there is a cat on the mat`
Score:	2 / 7

Brevity penalty

- r: sum of all minimal absolute length differences between candidates and referents.
- c: total length of all candidates
- BP: 1 if $c > r$ else $e^{1-\frac{r}{c}}$

BLEU

BP $\cdot$ the sum of weighted modified n-gram precision values for each n considered

BLEU scores: Properties

- Bounds: $[0, 1]$, with 1 the best, though with no expectation that any system will achieve 1.
- Value encoded:
  - Appropriate balance of (modified) precision and "recall" (BP).
  - Similar to word-error rate, but seeks to accommodate the fact that there are typically multiple suitable outputs for a given input.
- Weaknesses:
  - Callison-Burch et al. (2006) argue that BLEU fails to correlate with human scoring of translations.
  - Very sensitive to n-gram order.
  - Insensitive to n-gram types (*that dog* vs. *the dog* vs. *that toaster*).
  - Liu et al. (2016) specifically argue against BLEU as a metric for assessing dialogue systems.

Other reference-based metrics

Word-error rate	Edit-distance from a single reference text
BLEU	Modified precision and brevity penalty, against many reference texts
ROUGE	Recall-focused variant of BLEU, focused on assessing summarization systems
METEOR	Unigram-based alignments using exact match, stemming, synonyms
CIDEr	Weighted cosine similarity between TF-IDF vectors
BERTScore	Weighted MaxSim of token-level BERT representations

Image-based NLG metrics

- For the task of assessing texts associated with images, the reference-based metrics can be used if the needed human annotations exist.
- Reference-less metrics in this space seek to score text–image pairs with no need for human-created references:
  - CLIPScore (Hessel et al. 2021)
  - UMIC (Lee et al. 2021)
  - SPURTS (Feinglass and Yang 2021)
- Kreiss et al. (2022) criticize these methods as being insensitive to the context of the image and the purpose of the associated text, and they begin to design variants of CLIPScore that capture these dimensions of quality.

Task-oriented metrics

1. The classical off-the-shelf reference-based metrics will only capture aspects of the task to the extent that the human annotations do.
2. Model-based metrics could conceivably be tuned to specific tasks, but this is currently rare.
3. It is fruitful to think about what the goal of the generated tests is and consider whether one's evaluation could be based on that goal:
    - Can an agent that received the generated text use it to solve the task?
    - Was a specific piece of information reliably communicated?
    - Did the message lead the person to take a desirable action?

Newman et al. 2020

Datasets

Water and air for our field

Jacques Cousteau: *Water and air, the two essential fluids on which all life depends, have become global garbage cans.*

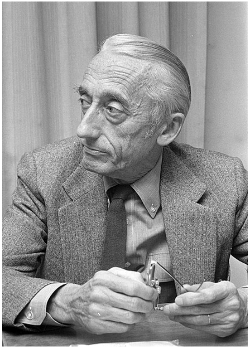
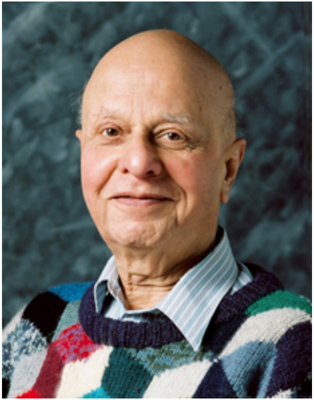


Photo credit: Wikipedia

We ask a lot of our datasets

1. Optimize models
2. Evaluate models
3. Compare models
4. Enable new capabilities in models
5. Measure fieldwide progress
6. Scientific inquiry

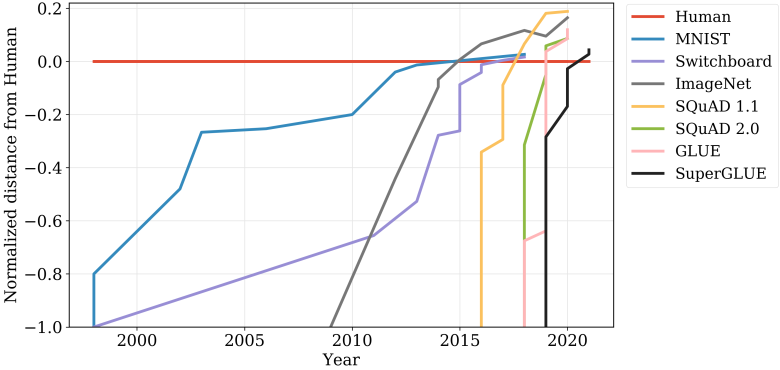
Seeing farther than ever before



Aravind Joshi: *Datasets as the telescopes of our field*

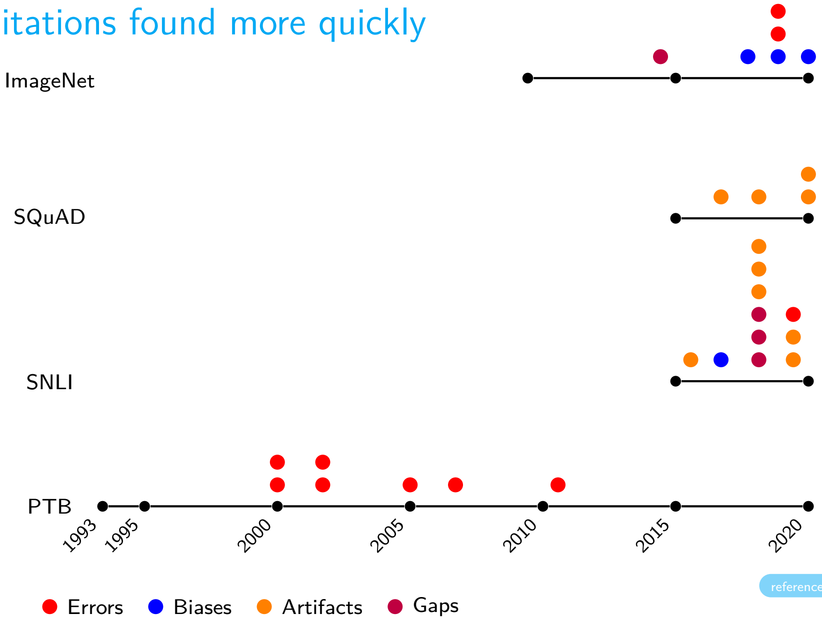
Photo credit: JoshiFest

Benchmarks saturate faster than ever



Kiela et al. 2021

Limitations found more quickly



references

Central questions

1. Naturalistic data or crowdsourcing? Both!
2. Adversarial examples or the most common cases? Both!
3. Synthetic or naturalistic benchmarks? Both!

Trade-offs

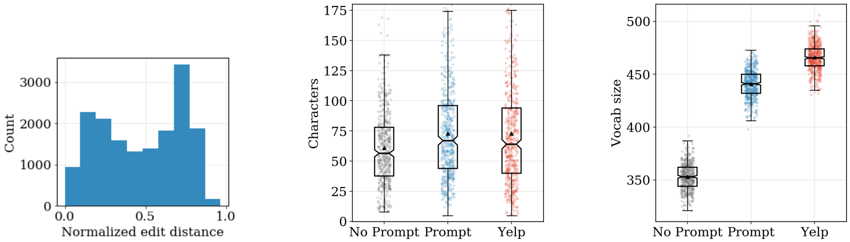
Naturalistic: Found and curated

- Abundant
- Inexpensive
- Genuine
- Uncontrolled
- Limited
- Intrusive

Crowdsourced: Lab-grown

- Controlled
- Privacy preserving
- Expressive
- Scarce
- Expensive
- Contrived

DynaSent: Prompts increase naturalism



1. "My sister hated the food but she's massively wrong."
2. "The cookies seemed dry to my boss but I couldn't disagree more."
3. "Breakfast is really good, if you're trying to feed it to dogs."

Potts et al. 2021; see also Bartolo et al. 2021

Adversarial examples or the most common cases?

Standard

Create a dataset from a single model-independent process and divide it into train/dev/test.

Adversarial assessment

A separate test set is created in ways that you suspect or know will be challenging given your system and/or the (Standard) train data.

Adversarial datasets

Dataset (train/dev/test) guided by attempts to fool a set of models.

Dynamics of adversarial datasets

1. SWAG to BERT to HellaSWAG (Zellers et al. 2018, 2019)
2. Adversarial NLI (Nie et al. 2020)
3. Beat the AI (Bartolo et al. 2020)
4. Dynabench Hate Speech (Vidgen et al. 2020)
5. DynaSent (Potts et al. 2021)

Counterpoint from Bowman and Dahl (2021)

Adversarial examples not a panacea

"Adversarial filtering [...] can systematically eliminate coverage of linguistic phenomena or skills that are necessary for the task but already well-solved by the adversary model. This mode-seeking (as opposed to mass covering) behavior by adversarial filtering, if left unchecked, tends to reduce dataset diversity and thus make validity harder to achieve."

Standard evaluations sufficient

"This position paper argues that concerns about standard benchmarks that motivate methods like adversarial filtering are justified, but that they can and should be addressed directly, and that it is possible and reasonable to do so in the context of static, IID evaluation."

The job to be done

0 "The food was good"

1 "My sister hated the food but she's massively wrong."

2 "The cookies seemed dry to my boss but I couldn't disagree more."

3 "Breakfast is really good, if you're trying to feed it to dogs."

4 "worthy of gasps of foodgasms"

Major lessons thus far

1. Top systems have often found *unsystematic* solutions.
2. Progress on challenge sets seems to correlate with meaningful progress.
3. Present-day systems get traction on adversarial cases without degradation on the general cases.
4. Adversarial examples often *define* public perception.

Synthetic or naturalistic benchmarks

Dataset and models as unknowns

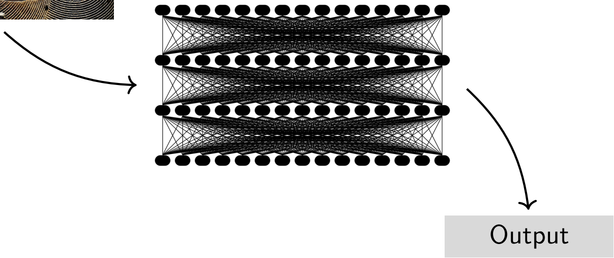


Photo credit: Robert Couse-Baker

Negation as a learning target

Intuitive learning target

If *A entails B* then *not-B entails not-A*

Observation

Top-performing NLI models fail to achieve the learning target (Yanaka et al. 2019, 2020; Hossain et al. 2020; Geiger et al. 2020).

Tempting conclusion

Top-performing models are incapable of learning negation.

Dataset observation

Negation is severely under-represented in NLI benchmarks.

MoNLI: A slightly synthetic dataset

Positive MoNLI (PMoNLI; 1,476 examples)

SNLI hypothesis (A)	Food was served.
WordNet	pizza $\sqsubset$ food
New example (B)	Pizza was served.
Positive MoNLI	(A) **neutral** (B)
Positive MoNLI	(B) **entailment** (A)

Negative MoNLI (NMoNLI; 1,202 examples)

SNLI hypothesis (A)	The children are **not** holding plants.
WordNet	flowers $\sqsubset$ plants
New example (B)	The children are **not** holding flowers.
Negative MoNLI	(A) **entailment** (B)
Negative MoNLI	(B) **neutral** (A)

Geiger et al. 2020

MoNLI as challenge dataset

			No MoNLI fine-tuning			With NMoNLI fine-tuning	
Model	Input pretrain	NLI train data	SNLI	PMoNLI	NMoNLI	SNLI	NMoNLI
BiLSTM	GloVe	SNLI train	81.6	73.2	37.9	74.6	93.5
ESIM	GloVe	SNLI train	87.9	86.6	39.4	56.9	96.2
BERT	BERT	SNLI train	90.8	94.4	2.2	90.5	90.0

Geiger et al. 2020

The value of messy data

When we turn to naturalistic data, we do so knowing:

1. that BERT can *in principle* learn negation; and
2. that data coverage will be a major factor.

Other vital issues for datasets

My central questions

1. Naturalistic data or crowdsourcing? Both!
2. Adversarial examples or the most common cases? Both!
3. Synthetic or naturalistic benchmarks? Both!

At least as important

1. Datasheets (Gebru et al. 2018)
2. Achieving cross-linguistic coverage for benchmarks
3. Statistical power (Bowman and Dahl 2021)
4. Pernicious social biases

Data organization

Train/Dev/Test

- Common in large publicly available datasets.
- Presupposes a fairly large dataset.
- We're all on the honor system to do test-set runs only when development is complete.
- The test part ensures consistent evaluations, but encourages hill climbing.

No fixed splits

- Small public datasets might not have predefined splits.
- A challenge for assessment: for robust comparisons, you really have to run all models using your assessment regime on your splits.
- For large datasets, you can impose splits and use them for the entire project:
  - Simplifies your experimental set-up.
  - Reduces hyperparameter optimization.
- For small datasets, imposing a split might leave too little data, leading to highly variable performance.

Cross-validation

In cross-validation, we take a set of examples and partition them into two or more train/test splits, and then we average over the results in some way.

Cross-validation: Random splits

Method

For k times:

1. Shuffle.
2. Split: t percent train, usually $1 - t$ test.
3. Conduct an evaluation.

In general (but not always), we want these splits to be *stratified* in the sense that the train and test splits have approximately the same distribution over the classes.

Trade-offs

- **Good**: you can create as many as you want without having this impact the ratio of training to testing examples.
- **Bad**: no guarantee that every example will be used the same number of times for training and testing.

```
from sklearn.model_selection import ShuffleSplit,
StratifiedShuffleSplit, train_test_split
```

Cross-validation: K-folds

Method

Splits
fold 1 fold 2 fold 3

Experiment 1	
Test	fold 1
Train	fold 2 fold 3

Experiment 2	
Test	fold 2
Train	fold 1 fold 3

Experiment 3	
Test	fold 3
Train	fold 1 fold 2

Trade-offs

- **Good**: every example appears in a train set exactly $k - 1$ times and in a test set exactly once.
- **Bad**: the size of k determines the size train/test:
  - 3-fold: train 67%, test 33%.
  - 10-fold: train 90%, test 10%.

```
from sklearn.model_selection import KFold,
StratifiedKFold, LeaveOneOut, cross_val_score
```

Model evaluation

Overview

- Baselines
- Hyperparameter optimization
- Classifier comparison
- Assessing models without convergence
- The role of random parameter initialization

Baselines

Evaluation numbers can never be understood properly in isolation:

1. Your system gets 0.95 F1! Is your task too easy?
2. Your system gets 0.60 F1. But what do humans get?

Baselines are crucial for strong experiments

- Defining baselines should not be an afterthought, but rather central to how you define your overall hypotheses.
- Baselines are essential to building a persuasive case.
- They can also be used to illuminate specific aspects of the problem and specific virtues of your proposed system.

Random baselines

Random baselines are almost always useful to include. sklearn:

- `DummyClassifier`
  - `stratified`
  - `uniform`
  - `most_frequent`
- `DummyRegressor`
  - `mean`
  - `median`

Task-specific baselines

It is worth considering whether your problem suggests a baseline that will reveal something about the problem or the ways it is modeled.

Two recent examples from NLU:

- NLI: Hypothesis-only baselines.
- The Story Cloze task: Distinguish between a coherent and incoherent ending for a story. Systems that look only at the ending options can do really well (Schwartz et al. 2017).

Hyperparameter optimization

Discussed in our unit on sentiment analysis. Rationales:

- Obtaining the best version of your model.
- Conducting fair comparisons between models.
- Understanding the stability of your architecture.

All hyperparameter tuning must be done only on train and development data.

The ideal hyperparameter optimization setting

1. For each hyperparameter, identify a large set of values for it.
2. Create a list of all the combinations of all the hyperparameter values. This will be the cross-product of all the values for all the features identified at step 1.
3. For each of the settings, cross-validate it on the available training data.
4. Choose the settings that did best in step 3, train on all the training data using those settings, and then evaluate that model on the test set.

An example

1. Parameter h_1 has 5 values.
2. Parameter h_2 has 10 values.
3. Total settings: $5 \cdot 10 = 50$.
4. Add h_3 with 2 values.
5. Total settings: $5 \cdot 10 \cdot 2 = 100$.
6. 5-fold cross-validation to select optimal parameters: 500 runs

Practical considerations

The above is untenable as a set of laws for the scientific community.

If we adopted it, then complex models trained on large datasets would end up disfavored, and only the very wealthy would be able to participate.

Rajkomar et al. (2018):

"the performance of all above neural networks were [sic] tuned automatically using Google Vizier [35] with a total of $> 201,000$ GPU hours"

Reasonable compromises

Pragmatic steps you can take to alleviate this problem, in descending order of attractiveness:

1. Random sampling and guided sampling allow you to explore a large space on a fixed budget of runs.
2. Search based on a few epochs of training. (Could be bolstered with short learning curves for different settings.)
3. Search based on subsets of the data. (However, some parameters will be very dependent on dataset size, so this can be risky.)
4. Via heuristic search, determine which hyperparameters matter less, and set them by hand. (Justify this in the paper!)
5. Find optimal hyperparameters via a single split and use them for all the subsequent splits. Justified if the splits are similar.
6. Adopt others' choices. The skeptic will complain that these findings don't translate to your new data sets, but it could be the only option.

Tools for hyperparameter search

- `from sklearn.model_selection import GridSearchCV, RandomizedSearchCV, HalvingGridSearchCV`
- scikit-optimize offers a variety of methods for guided search through the grid of hyperparameters.

Classifier comparison

Suppose you've assessed two classifier models. Their performance is probably different to some degree. What can be done to establish whether these models are different in any meaningful sense?

- Practical differences
- Confidence intervals
- Wilcoxon signed-rank test
- McNemar's test

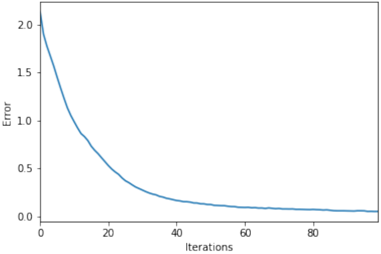
Assessing models without convergence

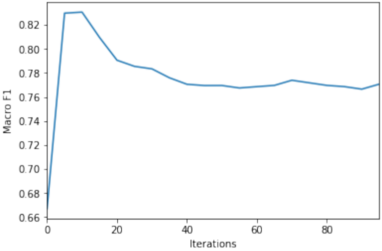
- When working with linear models, convergence issues rarely arise.
- With neural networks, convergence takes center stage:
  - The models rarely converge.
  - For they converge at different rates between runs.
  - Their performance on the test data is often heavily dependent on these differences.
- Sometimes a model with a low final error turns out to be great, and sometimes it turns out to be worse than one that finished with a higher error. Who knows?!

Incremental dev-set testing

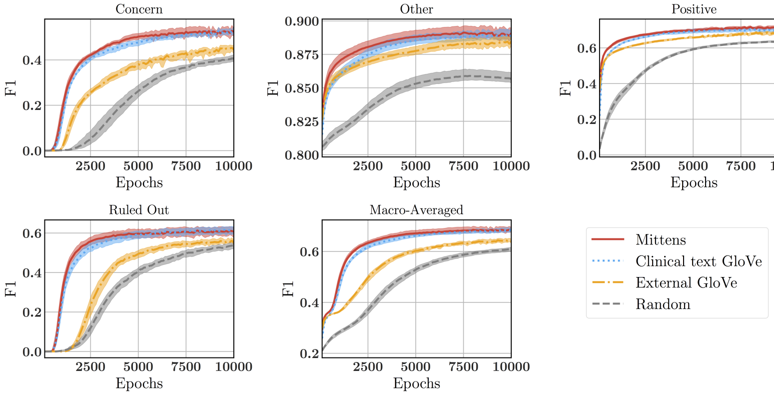
1. To address this uncertainty: regularly collect information about dev set performance as part of training.
2. For example, at every 100th iteration, one could make predictions on the dev set and store that vector of predictions.
3. All the PyTorch models for this course have an `early_stopping` with various controllable parameters.

A bit of motivation for early stopping





Learning curves with confidence intervals



Dingwall and Potts 2018

The role of random parameter initialization

1. Most deep learning models have their parameters initialized randomly
2. Clearly meaningful for non-convex optimization problems Simpler models can be impacted as well.
3. Reimers and Gurevych (2017):
   - Different initializations for neural sequence models can lead to statistically significant differences.
   - A number of recent systems are indistinguishable in terms of raw performance once this source of variation is taken into account.
4. Related: catastrophic failure as a result of unlucky initialization.
5. In `evaluation_methods.ipynb`: A feedforward network on the XOR problem succeeds 8 of 10 times.

Conclusion

Experiment protocols

This is a short, structured report designed to help you establish your core experimental framework. The required sections are as follows:

1. Hypotheses
2. Data
3. Metrics
4. Models
5. General reasoning
6. Summary of progress so far
7. References section

Goal: clarity of project goals, identification of obstacles and project risks.

An ideal moment for innovation

1. Architecture innovation – overrated!
2. Metric innovation – way underrated!
3. Evaluation innovation – way underrated!
4. Task innovation – underrated!
5. Exhaustive hyperparameter search – needs to be weighed against other factors!

References I

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Max Bartolo, Tristan Thrush, Sebastian Riedel, Pontus Stenetorp, Robin Jia, and Douwe Kiela. 2021. Models in the loop: Aiding crowdworkers with generative annotation assistants. *CoRR*, abs/2112.09062.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Abeba Birhane, Pratyusha Kalluri, Dallas Card, William Agnew, Ravit Dotan, and Michelle Bao. 2021. The values encoded in machine learning research. *arXiv preprint arXiv:2106.15590*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Stroudsburg, PA. Association for Computational Linguistics.

Samuel R Bowman and George E Dahl. 2021. What will it take to fix benchmarking in natural language understanding? *arXiv preprint arXiv:2104.02145*.

Adriane Boyd, Markus Dickinson, and Detmar Meurers. 2008. On detecting errors in dependency treebanks. *Research on Language and Computation*, 6(2):113–137.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A Smith. 2021. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7282–7296, Online. Association for Computational Linguistics.

Cody Coleman, Deepak Narayanan, Daniel Kang, Tian Zhao, Jian Zhang, Luigi Nardi, Peter Bailis, Kunle Olukotun, Chris Ré, and Matei Zaharia. 2017. Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101):102.

Kate Crawford and Trevor Paglen. 2021. Excavating ai: The politics of images in machine learning training sets. *AI & SOCIETY*, pages 1–12.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S Bernstein, Alex Berg, and Li Fei-Fei. 2014. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102.

Markus Dickinson and W. Detmar Meurers. 2003a. Detecting errors in part-of-speech annotation. In *10th Conference of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary. Association for Computational Linguistics.

References II

Markus Dickinson and W. Detmar Meurers. 2005. Detecting errors in discontinuous structural annotation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 322–329, Ann Arbor, Michigan. Association for Computational Linguistics.

Markus Dickinson and Walt Detmar Meurers. 2003b. Detecting inconsistencies in treebanks. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories*.

Nicholas Dingwall and Christopher Potts. 2018. Mittens: An extension of GloVe for learning domain-specialized representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 212–217, Stroudsburg, PA. Association for Computational Linguistics.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Eleazar Eskin. 2000. Detecting errors within a corpus using anomaly detection. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of NLP leaderboards. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4846–4853, Online. Association for Computational Linguistics.

Joshua Feinglass and Yezhou Yang. 2021. SMURF: SeMantic and linguistic UndeRstanding fusion for caption evaluation via typicality analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2250–2260, Online. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumeé III, and Kate Crawford. 2018. Datasheets for datasets. *arXiv preprint arXiv:1803.09010*.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

References III

Hans van Halteren. 2000. The detection of inconsistency in manually tagged text. In *Proceedings of the COLING-2000 Workshop on Linguistically Interpreted Corpora*, pages 48–55, Centre Universitaire, Luxembourg. International Committee on Computational Linguistics.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Md Mosharaf Hossain, Venelin Kovatchev, Pranoy Dutta, Tiffany Kao, Elizabeth Wei, and Eduardo Blanco. 2020. An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9106–9118, Online. Association for Computational Linguistics.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Elisa Kreiss, Cynthia Bennett, Shayan Hooshmand, Eric Zelikman, Meredith Ringel Morris, and Christopher Potts. 2022. Context matters for image descriptions for accessibility: Challenges for referenceless evaluation metrics. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4685–4697, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Trung Bui, and Kyomin Jung. 2021. UMIC: An unreferenced metric for image captioning via contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 220–226, Online. Association for Computational Linguistics.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Pengfei Liu, Jinlan Fu, Yang Xiao, Weizhe Yuan, Shuaicheng Chang, Junqi Dai, Yixin Liu, Zihuiwen Ye, and Graham Neubig. 2021. Explainaboard: An explainable leaderboard for NLP. *arXiv preprint arXiv:2104.06387*.

Zhiyi Ma, Kawin Ethayarajh, Tristan Thrush, Somya Jain, Ledell Wu, Robin Jia, Christopher Potts, Adina Williams, and Douwe Kiela. 2021. Dynaboard: An evaluation-as-a-service platform for holistic next-generation benchmarking. In *Advances in Neural Information Processing Systems*, volume 34, pages 10351–10367.

References IV

Christopher D. Manning. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing – Part I*, number 6608 in Lecture Notes in Computer Science, pages 171–189. Springer, Berlin.

Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Benjamin Newman, Reuben Cohn-Gordon, and Christopher Potts. 2020. Communication-based evaluation for natural language generation. In *Proceedings of the Society for Computation in Linguistics*, pages 234–244, Washington, D.C. Linguistic Society of America.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. 2021. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

Christopher Potts, Zhengxuan Wu, Atticus Geiger, and Douwe Kiela. 2021. DynaSent: A dynamic benchmark for sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2388–2404, Online. Association for Computational Linguistics.

Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Peter J Liu, Xiaobing Liu, Mimi Sun, Patrik Sundberg, Hector Yee, et al. 2018. Scalable and accurate deep learning for electronic health records. *arXiv preprint arXiv:1801.07860*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

References V

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400, Long Beach, California, USA. PMLR.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Philip Resnik and Jimmy Lin. 2010. Evaluation of NLP systems. In Alexander Clark, Chris Fox, and Shalom Lappin, editors, *The Handbook of Computational Linguistics and Natural Language Processing*, pages 271–295. Wiley-Blackwell.

Rachel Rudinger, Chandler May, and Benjamin Van Durme. 2017. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 74–79, Valencia, Spain. Association for Computational Linguistics.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. Story cloze task: UW NLP system. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 52–55, Valencia, Spain. Association for Computational Linguistics.

Stuart Shieber. 1994. Lessons from a restricted Turing test. *Communications of the ACM*, 37(6):70–78.

Vincent Sitzmann, Martina Marek, and Leonid Keselman. 2016. Multimodal natural language inference. Final paper, CS224u, Stanford University.

Noah A. Smith. 2011. *Linguistic Structure Prediction*. Morgan & Claypool, San Rafael, CA.

Pierre Stock and Moustapha Cisse. 2018. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 498–512.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.

Masatoshi Tsuchiya. 2018. Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2020. Learning from the worst: Dynamically generated datasets to improve online hate detection. *arXiv prerint arXiv:2012.15761*.

Dirk Weissenborn, Georg Wiese, and Laura Seiffe. 2017. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117, Online. Association for Computational Linguistics.

References VI

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019. HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 250–255, Minneapolis, Minnesota. Association for Computational Linguistics.

Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. 2020. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 547–558.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

References for the benchmark timeline

Penn Treebank (Marcus et al. 1994)

1.	van Halteren 2000	E
2.	Eskin 2000	E
3.	Dickinson and Meurers 2003a	E
4.	Dickinson and Meurers 2003b	E
5.	Dickinson and Meurers 2005	E
6.	Boyd et al. 2008	E
7.	Manning 2011	E

SNLI (Bowman et al. 2015)

1.	Sitzmann et al. 2016	A
2.	Rudinger et al. 2017	S
3.	Naik et al. 2018	G
4.	Glockner et al. 2018	G
5.	Naik et al. 2018	G
6.	Poliak et al. 2018	A
7.	Tsuchiya 2018	A
8.	Gururangan et al. 2018	A
9.	Belinkov et al. 2019	A
10.	McCoy et al. 2019	A

SQuAD (Rajpurkar et al. 2016, 2018)

1.	Weissenborn et al. 2017	A
2.	Sugawara et al. 2018	A
3.	Bartolo et al. 2020	A
4.	Lewis et al. 2021	A

ImageNet (Deng et al. 2009)

1.	Deng et al. 2014	G
2.	Stock and Cisse 2018	B
3.	Yang et al. 2020	B
4.	Recht et al. 2019	E
5.	Northcutt et al. 2021	E
6.	Crawford and Paglen 2021	B

timeline slide