CS388: Natural Language Processing
Lecture 10: Evaluation Principles, Dataset Artifacts

Greg Durrett

The University of Texas at Austin
Announcements

› Final project proposals due next Tuesday

› P3 released next week
Recap

▪ **Pretraining (BERT):**
  ▪ Train a big model to fill in masked-out words, then adapt it to other tasks. Led to big gains in **question answering** and **NLI** performance. BART/T5, GPT-3, etc. push this further and extend it to other tasks

▪ **Decoding methods:** nucleus sampling > greedy for open-ended tasks

▪ **Two tasks we’ll focus on today:** Question answering (QA)...
  ▪ “What was Marie Curie the first female recipient of?”
    -> “The Nobel Prize” (find this span in a document)

▪ ...and **NLI**
  ▪ "But I thought you'd sworn off coffee." \(contradicts\) "I thought that you vowed to drink more coffee."
Today

- Evaluation in NLP: benchmarks and generalization
- Spurious correlations / dataset artifacts
- Debiasing
Cross-Dataset Evaluation
Principles of Evaluation Suites

- Training and testing on i.i.d. data with big neural models often yields very high performance

- “Solving” a task (getting human-level performance) may be useful, but often can’t tell us about our models more broadly

- A parable of single-task evaluation: SWAG
On stage, a woman takes a seat at the piano. She
a) sits on a bench as her sister plays with the doll.
b) smiles with someone as the music plays.
c) is in the crowd, watching the dancers.
d) nervously sets her fingers on the keys.

- Text-only data comes from video captions in ActivityNet
- **Adversarial filtering** to produce the negative multiple choice answers
- They said models get ~60% and humans get ~85%, but BERT immediately solved this dataset when it was released

```
while convergence not reached do
  • Split the dataset $\mathcal{D}$ randomly up into training and testing portions $\mathcal{D}^{tr}$ and $\mathcal{D}^{te}$.
  • Optimize a model $f_\theta$ on $\mathcal{D}^{tr}$.
  for index $i$ in $\mathcal{D}^{te}$ do
    • Identify easy indices:
      $A_i^{easy} = \{ j \in A_i : f_\theta(x_i^+) > f_\theta(x_i,j) \}$
    • Replace $N^{easy}$ easy indices $j \in A_i^{easy}$ with adversarial indices $k \notin A_i$ satisfying $f_\theta(x_i^+,k) > f_\theta(x_i,j)$.
  end for
end while
```
Intuition

General drop-off in how many hard examples there are

Example difficulty
Intuition

If you exclude easy examples, most of the remaining examples are just slightly harder than what you excluded.

General drop-off in how many hard examples there are.
Principles of Evaluation Suites

- Training and testing on i.i.d. data with big neural models often yields very high performance

- “Solving” a task (getting human-level performance) may be useful, but often can’t tell us about our models more broadly

- Designing a single, difficult task is really challenging!

- To assess big models, we need evaluation suites (benchmarks) like GLUE

- What makes a good evaluation suite of tasks?
Principles of Evaluation Suites

- Difficulty: even if some task can be solved by hand-engineering, it should be hard to solve all $N$ tasks

- Diverse: doing well on it should say something useful

- Good “yardstick”: should understand where human performance is and what good performance on the task would mean

- GLUE was the first of these, but it wasn’t really diverse and it was too easy. Next step: SuperGLUE

Alex Wang et al., 2019
SuperGLUE: Task Requirements

- Task substance: “Tasks should test a system’s ability to understand and reason about texts in English.”

- Task difficulty: “Tasks should be beyond the scope of current state-of-the-art systems, but solvable by most college-educated English speakers.” (notably they excluded domain-specific tasks, which have become more popular these days, e.g., the bar exam)

- Evaluable: this is challenging to find!

- Public dataset, good license, etc.

Alex Wang et al., 2019
### SuperGLUE: Performance

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Most Frequent</td>
<td>47.1</td>
<td>62.3</td>
<td>21.7/48.4</td>
<td>50.0</td>
<td>61.1 / 0.3</td>
<td>33.4/32.5</td>
<td>50.3</td>
<td>50.0</td>
<td>65.1</td>
<td>0.0</td>
<td>100.0/ 50.0</td>
<td></td>
</tr>
<tr>
<td>CBoW</td>
<td>44.3</td>
<td>62.1</td>
<td>49.0/71.2</td>
<td>51.6</td>
<td>0.0 / 0.4</td>
<td>14.0/13.6</td>
<td>49.7</td>
<td>53.0</td>
<td>65.1</td>
<td>-0.4</td>
<td>100.0/ 50.0</td>
<td></td>
</tr>
<tr>
<td>BERT</td>
<td>69.0</td>
<td>77.4</td>
<td>75.7/83.6</td>
<td>70.6</td>
<td>70.0 / 24.0</td>
<td>72.0/71.3</td>
<td>71.6</td>
<td>69.5</td>
<td>64.3</td>
<td>23.0</td>
<td>97.8 / 51.7</td>
<td></td>
</tr>
<tr>
<td>BERT++</td>
<td>71.5</td>
<td>79.0</td>
<td><strong>84.7/90.4</strong></td>
<td>73.8</td>
<td>70.0 / 24.1</td>
<td>72.0/71.3</td>
<td>79.0</td>
<td><strong>69.5</strong></td>
<td><strong>64.3</strong></td>
<td>38.0</td>
<td>99.4 / 51.4</td>
<td></td>
</tr>
<tr>
<td>Outside Best</td>
<td>-</td>
<td><strong>80.4</strong></td>
<td>-/-</td>
<td><strong>84.4</strong></td>
<td><strong>70.4</strong>/<em>24.5</em>/</td>
<td><strong>74.8/73.0</strong></td>
<td><strong>82.7</strong></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>/-</td>
</tr>
<tr>
<td>Human (est.)</td>
<td>89.8</td>
<td>89.0</td>
<td>95.8/98.9</td>
<td>100.0</td>
<td>81.8*/51.9*</td>
<td>91.7/91.3</td>
<td>93.6</td>
<td>80.0</td>
<td>100.0</td>
<td>77.0</td>
<td><strong>99.3/99.7</strong></td>
<td></td>
</tr>
</tbody>
</table>

- RoBERTa in 2019: 84.6
- DeBERTa in 2020: 90.3. **Even SuperGLUE was solved quickly!**

Alex Wang et al., 2019
SuperGLUE: Performance

As reported in BIGBench:

SuperGLUE state of the art over time

![Graph showing SuperGLUE performance over time]
BIG-bench

- 204 tasks, 444 authors
BIG-bench

- “Beyond the Imitation Game” — aim to learn more than what’s possible from model vs. human performance

- Particular emphasis on scaling

- Primarily for pre-trained models without fine-tuning. Therefore, not all tasks have large training (or even test!) sets
MMLU task (Hendrycks et al., 2020): 57 high school/college/professional exams:

- **Conceptual Physics**: When you drop a ball from rest it accelerates downward at 9.8 m/s². If you instead throw it downward assuming no air resistance its acceleration immediately after leaving your hand is
  (A) 9.8 m/s² ✓
  (B) more than 9.8 m/s² ✗
  (C) less than 9.8 m/s² ✗
  (D) Cannot say unless the speed of throw is given. ✗

- **College Mathematics**: In the complex z-plane, the set of points satisfying the equation $z^2 = |z|^2$ is a
  (A) pair of points ✗
  (B) circle ✗
  (C) half-line ✗
  (D) line ✓

Figure 4: Examples from the Conceptual Physics and College Mathematics STEM tasks.
Chung et al. (2022)
MMLU

MMLU task (Hendrycks et al., 2020): 57 high school/college/professional exams:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Random</td>
<td>25.0</td>
</tr>
<tr>
<td></td>
<td>Average human rater</td>
<td>34.5</td>
</tr>
<tr>
<td>May 2020</td>
<td>GPT-3 5-shot</td>
<td>43.9</td>
</tr>
<tr>
<td>Mar. 2022</td>
<td>Chinchilla 5-shot</td>
<td>67.6</td>
</tr>
<tr>
<td>Apr. 2022</td>
<td>PaLM 5-shot</td>
<td>69.3</td>
</tr>
<tr>
<td>Oct. 2022</td>
<td>Flan-PaLM 5-shot</td>
<td>72.2</td>
</tr>
<tr>
<td>Oct. 2022</td>
<td>Flan-PaLM 5-shot: CoT + SC</td>
<td>75.2</td>
</tr>
<tr>
<td></td>
<td>Average human expert</td>
<td>89.8</td>
</tr>
</tbody>
</table>

Chung et al. (2022)
Chung et al. (2022)

**MMLU**

<table>
<thead>
<tr>
<th>Model</th>
<th>Finetuning Mixtures</th>
<th>Tasks</th>
<th>Norm. avg.</th>
<th>MMLU</th>
<th>BBH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Direct</td>
<td>CoT</td>
</tr>
<tr>
<td>540B</td>
<td>None (no finetuning)</td>
<td>0</td>
<td>49.1</td>
<td>71.3</td>
<td>62.9</td>
</tr>
<tr>
<td>CoT</td>
<td></td>
<td>9</td>
<td>52.6 (+3.5)</td>
<td>68.8</td>
<td>64.8</td>
</tr>
<tr>
<td>CoT, Muffin</td>
<td></td>
<td>89</td>
<td>57.0 (+7.9)</td>
<td>71.8</td>
<td>66.7</td>
</tr>
<tr>
<td>CoT, Muffin, T0-SF</td>
<td></td>
<td>282</td>
<td>57.5 (+8.4)</td>
<td>72.9</td>
<td><strong>68.2</strong></td>
</tr>
<tr>
<td>CoT, Muffin, T0-SF, NIV2</td>
<td></td>
<td>1,836</td>
<td><strong>58.5 (+9.4)</strong></td>
<td><strong>73.2</strong></td>
<td>68.1</td>
</tr>
</tbody>
</table>

- Human performance estimates are ~90 on MMLU, ~80 on Big-Bench (BBH). Even getting close on these tasks!
Evaluation Under Distribution Shift
Model Performance

- If models can be fine-tuned on each of n tasks in an evaluation suite and perform very well on the held-out test dataset, have we solved everything we want?

- What can go wrong?
Generalization

- If a model does well on train but poorly on test data, it *doesn’t generalize*
- A model can do well on its test data and still fail to generalize *out of distribution* — arguably an even more important notion
- Many notions of generalization. Example: POS tagging

![Diagram showing generalization的概念](attachment:diagram.png)

- Train data: English, Wall Street Journal
- Test data: English, also WSJ
- Other domains, languages, etc.
  - Easy: English Tweets
  - Hard: English fiction
  - Other: French newswire

(d pageable with multilingual models)
Generalization: QA

Train data
- SQuAD: factoid questions with answers on Wikipedia

Test data
- SQuAD

Other domains
- Science questions
- Unanswerable questions
- French questions

Other types of reasoning, such as *multi-hop questions*

Who won the Nobel in Chemistry the year Marie Curie won the Nobel in Physics?
Generalization

- Just doing well on a single test set is not that useful

- We want POS taggers, QA systems, and more that can generalize to new settings so we can deploy them in practice

- Sometimes, you can get very good test performance but the model generalizes very poorly. How does this happen?
Annotation Artifacts, Reasoning Shortcuts: QA
Some datasets might be easy because of how they’re constructed, especially in QA and NLI

What becomes of Macbeth?

What does Macduff do to Macbeth?

What violent act does Macduff perform upon Macbeth?

All questions have the same answer. But some are more easily guessable
Reminder: QA with BERT

Devlin et al. (2019)
What degree did Martin Luther receive on October 19, 1512?

On October 19, 1512, Luther was awarded his doctorate of theology and, on October 21, 1512, was received into the senate of the theological faculty of the University of Wittenberg. He spent the rest of his career in this position at the University of Wittenberg.

- What should the model be doing? Corresponding Martin Luther with Luther, matching October 19, 1512 between question and passage
What degree did Martin Luther receive?

On October 19, 1512, Luther was awarded his doctorate of theology and, on October 21, 1512, was received into the senate of the theological faculty of the University of Wittenberg. He spent the rest of his career in this position at the University of Wittenberg.

‣ Only one possible degree here! Model only needs to see “what degree” and will not learn to use the rest of the context!
QA: Answer Type Heuristics

- Question type is powerful indicator. Only a couple of locations in this context!

Where ____?

On October 19, 1512, Luther was awarded his doctorate of theology and, on October 21, 1512, was received into the senate of the theological faculty of the University of Wittenberg. He spent the rest of his career in this position at the University of Wittenberg.

Who ____?

When ____?
QA: Answer Type Heuristics

- Question type is powerful indicator. Only a couple of locations in this context!

Where ____? Who ____? When ____?

On October 19, 1512, Luther was awarded his doctorate of theology and, on October 21, 1512, was received into the senate of the theological faculty of the University of Wittenberg. He spent the rest of his career in this position at the University of Wittenberg.

- What will happen if we train on this data?
  - Will loss decrease?
  - How will the model learn to “behave”?
Annotation Artifacts,
Reasoning Shortcuts: NLI
Reminder: NLI with BERT

entailed/neutral/contradiction

Devlin et al. (2019)
NLI: Hypothesis-only Baselines

Premise: A woman on a deck is selling bamboo sticks.

Hypothesis: A man is selling bamboo sticks

Hypothesis: A man is juggling flaming chainsaws

Hypothesis: Eighteen flying monkeys are in outer space

- Not all of these things have the same likelihood of being true a priori
- What might the model learn to do in this case?
### NLI: Hypothesis-only Baselines

<table>
<thead>
<tr>
<th>Premise</th>
<th>A woman selling bamboo sticks talking to two men on a loading dock.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Entailment</strong></td>
<td>There are <strong>at least</strong> three <strong>people</strong> on a loading dock.</td>
</tr>
<tr>
<td><strong>Neutral</strong></td>
<td>A woman is selling bamboo sticks <strong>to help provide for her family.</strong></td>
</tr>
<tr>
<td><strong>Contradiction</strong></td>
<td>A woman is <strong>not</strong> taking money for any of her sticks.</td>
</tr>
</tbody>
</table>

- What’s different about this neutral sentence?
  - To create neutral sentences: annotators **add information**

- What’s different about this contradictory sentence?
  - To create contradictions: annotators **add negation**

- These are not broadly representative of what can happen in other settings. There is no “natural” distribution of NLI, but this is still very restrictive
NLI: Hypothesis-only Baselines

<table>
<thead>
<tr>
<th>Premise</th>
<th>A woman selling bamboo sticks talking to two men on a loading dock.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entailment</td>
<td>There are <strong>at least</strong> three <strong>people</strong> on a loading dock.</td>
</tr>
<tr>
<td>Neutral</td>
<td>A woman is selling bamboo sticks <strong>to help provide for her family.</strong></td>
</tr>
<tr>
<td>Contradiction</td>
<td>A woman is <strong>not</strong> taking money for any of her sticks.</td>
</tr>
</tbody>
</table>

- Models can detect new information or negation easily
- Models can do very well **without looking at the premise**

<table>
<thead>
<tr>
<th>Hyp-only model</th>
<th>Majority class</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNLI</td>
<td>69.17</td>
<td>33.82</td>
</tr>
<tr>
<td>MNLI-1</td>
<td>55.52</td>
<td>35.45</td>
</tr>
<tr>
<td>MNLI-2</td>
<td>55.18</td>
<td>35.22</td>
</tr>
</tbody>
</table>

Gururangan et al. (2018); Poliak et al. (2018)
<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
</table>
| Lexical overlap | Assume that a premise entails all hypotheses constructed from words in the premise | **The doctor** was paid by the actor.  
                                  → The doctor paid the actor.  
                                  **WRONG**                                                                 |
| Subsequence     | Assume that a premise entails all of its contiguous subsequences.          | The doctor near **the actor** danced.  
                                  → The actor danced.  
                                  **WRONG**                                                                 |
| Constituent     | Assume that a premise entails all complete subtrees in its parse tree.     | **If the artist slept**, the actor ran.  
                                  → The artist slept.  
                                  **WRONG**                                                                 |

- Word overlap supersedes actual reasoning in these cases
- They create a test set (HANS) consisting of cases where heuristics like word overlap are misleading. Very low performance

McCoy et al. (2019)
Evidence of Spurious Correlations: Contrast Sets

‣ How do we control for annotation artifacts? Things like “premises and hypotheses overlap too much” aren’t easy to see!

‣ For any particular effect like lexical overlap, we could try to annotate data that “breaks” that effect

‣ Issue: breaking one correlation may just result in another one surfacing. How do we “break” them all at the same time?

‣ Solution: construct new examples through *minimal edits that change the label*.

Gardner et al. (2020)
Evidence of Spurious Correlations: Contrast Sets

Hardly one to be faulted for his ambition or his vision, it is genuinely unexpected, then, to see all Park’s effort add up to so very little. ... The premise is promising, gags are copious and offbeat humour abounds but it all fails miserably to create any meaningful connection with the audience. *(Label: Negative)*

By minimally editing an example, we control for pretty much all of the possible shortcuts that apply to the original.

- E.g., [summary starts with “Hardly” -> negative] is a pattern that could not hold anymore

Hardly one to be faulted for his ambition or his vision, *here we see all Park’s effort come to fruition.* ... The premise is *perfect*, gags are *hilarious* and offbeat humour abounds, *and it creates a deep* connection with the audience. *(Label: Positive)*

Gardner et al. (2020)
## Evidence of Spurious Correlations: Contrast Sets

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Examples</th>
<th># Sets</th>
<th>Model</th>
<th>Original Test</th>
<th>Contrast</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLVR2</td>
<td>994</td>
<td>479</td>
<td>LXMERT</td>
<td>76.4</td>
<td>61.1</td>
</tr>
<tr>
<td>IMDb</td>
<td>488</td>
<td>488</td>
<td>BERT</td>
<td>93.8</td>
<td>84.2</td>
</tr>
<tr>
<td>MATRES</td>
<td>401</td>
<td>239</td>
<td>CogCompTime2.0</td>
<td>73.2</td>
<td>63.3</td>
</tr>
<tr>
<td>UD English</td>
<td>150</td>
<td>150</td>
<td>Biaffine + ELMo</td>
<td>64.7</td>
<td>46.0</td>
</tr>
<tr>
<td>PERSPECTRUM</td>
<td>217</td>
<td>217</td>
<td>RoBERTa</td>
<td>90.3</td>
<td>85.7</td>
</tr>
<tr>
<td>DROP</td>
<td>947</td>
<td>623</td>
<td>MTMSN</td>
<td>79.9</td>
<td>54.2</td>
</tr>
</tbody>
</table>
Solutions
Broad Solutions

- Most solutions involve changing what data is trained on
  - Subset of data
  - Soft subset (i.e., reweight the existing examples)
  - Superset: add adversarially-constructed data, contrast sets, etc.
- For subsets: what do we train on?
  - Don’t train on stuff that allows you to cheat
  - Train on examples that teach the real task rather than shortcuts
What happens with each particular example during training?

Spurious correlations are *easy to learn*: a model should learn these early and always get them right.

Imagine a very challenging example

- Model prediction may change a lot as it learns this example, may be variable in its predictions.

Imagine a mislabeled example

- Probably just always wrong unless it gets overfit.
Data Maps

- Confidence: mean probability of correct label
- Variability: standard deviation in probability of the correct label
- Ambiguous examples: possible learnable (model knows it sometimes but not other times), but hard!
Data Maps

› What to do with them?

› Training on hard-to-learn or ambiguous examples leads to better performance out-of-domain

Swayamdipta et al. (2021)
Debiasing

- Other ways to identify easy examples other than data maps

- Train some kind of a weak model and discount examples that it fits easily

\[
\mathcal{L}(\theta_d) = -(1 - p_{b(i,c)}^{(i,c)}) y^{(i)} \cdot \log p_d
\]

probability under a copy of the model trained for a few epochs on a small subset of data (bad model)

Utama et al. (2020)
Debiasing

<table>
<thead>
<tr>
<th>Method</th>
<th>MNLI (Acc.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev</td>
</tr>
<tr>
<td>BERT-base</td>
<td>84.5</td>
</tr>
<tr>
<td>Reweighting$_{\text{known-bias}}$</td>
<td>83.5$^\dagger$</td>
</tr>
<tr>
<td>Reweighting$_{\text{self-debias}}$</td>
<td>81.4</td>
</tr>
<tr>
<td>Reweighting $\blackspade_{\text{self-debias}}$</td>
<td>82.3</td>
</tr>
</tbody>
</table>

- On the challenging HANS test set for NLI, this debiasing improves performance substantially.
- In-domain MNLI performance goes down.

Utama et al. (2020)
Debiasing

- Other work has explored similar approaches using a known bias model

\[ \hat{p}_i = \text{softmax}(\log(p_i) + \log(b_i)) \]

Probabilities from learned bias model — like the weak model from Utama et al. (prev. slides), but you define its structure

- *Ensembles* the weak model with the model you actually learn.
- Your actual model learns the *residuals* of the weak model: the difference between the weak model's output distribution and the target distribution.
- This lets it avoid learning the weak model's biases!
Takeaways

‣ Strong neural models trained on “tough” datasets may fail to generalize because they learn annotation artifacts

‣ By reweighting data or changing the training paradigm, you can learn a model that generalizes better

‣ Most gains will show up \textbf{out-of-domain}. Very hard to get substantial improvements on the same dataset, unless you consider small subsets of examples (e.g., the toughest 1% of examples by some measure)

‣ As more “generalist” LLMs are learned, this problem goes away...but there’s always a tradeoff when you want to fine-tune them for certain tasks

‣ Next time: further understanding in-context learning