bibinlp.umiacs.umd.edu

Hal Daumé III, Sudha Rao, Allyson Ettinger
Ephraim Rothschild & Harita Kannan
Emily Bender (UW)

Photo credit:
Pavel Blazek

Motivation

- Panel at Representation Learning in NLP

We bet it would be pretty easy to construct inputs to NLP systems that make them fail dramatically!

Motivation

Inspiration: builditbreakit.org

Computer Science > Cryptography and Security

Build It, Break It, Fix It: Contesting Secure Development

Andrew Ruef, Michael Hicks, James Parker, Dave Levin, Michelle L. Mazurek, Piotr Mardziel

(Submitted on 6 Jun 2016 (v1), last revised 19 Aug 2016 (this version, v2))

Typical security contests focus on breaking or mitigating the impact of buggy systems. We present the Build-it Break-it Fix-it BIBIFI contest which aims to assess the ability to securely build software not just break it. In BIBIFI teams build specified software with the goal of maximizing correctness performance and security. The latter is tested when teams attempt to break other teams submissions. Winners are chosen from among the best builders and the best breakers. BIBIFI was designed to be open-ended - teams can use any language tool process etc. that they like. As such contest outcomes shed light on factors that correlate with successfully building secure software and breaking insecure software. During we ran three contests involving a total of teams and two different programming problems. Quantitative analysis from these contests found that the most efficient build-it submissions used CC but submissions coded in a statically-typed language were less likely to have a security flaw build-it teams with diverse programming-language knowledge also produced more secure code. Shorter programs correlated with better scores. Break-it teams that were also build-it teams were significantly better at finding security bugs.

Subjects: **Cryptography and Security (cs.CR)**; Software Engineering (cs.SE)

Cite as: **arXiv:1606.01881 [cs.CR]**

(or **arXiv:1606.01881v2 [cs.CR]** for this version)

[build it | break it] nlp: goals

- “if it's rare it doesn't matter”
  - move beyond PAC-style model of NLP
  - enable better error analysis
  - make NLP systems fail-safe/fail-soft
- draw together NLP and linguistics

shared task setup

- Building round
  - we give out training data
  - builders build a system
  - builders give predictions on blind dev data
- Breaking round
  - breakers construct examples to break systems
  - examples are in the form of minimal pairs
- Judgment round
  - builders run their systems on breaker-data
  - points for robust systems & good breakers

how does breaking work?

- We give you *blind test data*
  - a collection of unlabeled examples
- You must generate minimal pairs
  - Pair is example of the form (a,b)
  - Where a is drawn from the *blind test data*
  - And b is a “small edit” of a
- You must also provide labels for minimal pairs
  - Labels may, but don't have to, disagree

breaking example (sentiment analysis)

- Blind test data contains, eg:
  - ? Every actor in this movie is horrible.
  - ? I love this movie!
- You can generate easy minimal pair:
  - -1 Every actor in this movie is horrible.
  - +1 Every actor in this movie is wonderful.
- This is probably not so useful
  - most builder-systems are likely to get this right

breaking example (sentiment analysis)

- Blind test data contains, eg:
  - ? Every actor in this movie is horrible.
  - ? I love this movie!
- You can generate harder minimal pair:
  - +1 I love this movie!
  - +1 I am mad for this movie!
- This is plausibly better
  - "mad for" might mislead systems

two tasks

- Sentiment analysis
  - Data from Pang+Lee
  - Rotten Tomatoes reviews
  - Low barrier to entry
  - Less linguistically interesting

- Semantic role labeling as question answering
  - Data/task from He+Lewis+Zettlemoyer
  - Higher barrier to entry
  - More linguistically interesting

semantic role labeling as qa

dada.cs.washington.edu/qasrl

UCD finished the 2006 championship as Dublin champions, by beating St Vincents in the final.

- Who **finished** something? UCD
- What did someone **finish**? the 2006 championship
- What did someone **finish** something as? Dublin champions
- How did someone **finish** something? by beating St Vincents in the final

- Who **beat** someone? UCD
- When did someone **beat** someone? in the final
- Who did someone **beat**? St Vincents

scoring for builders (work in progress)

- accuracy on breaker-tests
- (lack of) degradation on second example in minimal pair
  - (ie if you didn't get the first half right, you pay less for getting the second half wrong)
- number of distinct breakers that you're robust to
- allow abstentions (probably future work)

scoring for breakers (work in progress)

- Maximal credit when systems succeed for a but not for b in minimal pair (or vice versa?)
- Score based on differentiation between systems
  - if all systems break on your input → less good
  - if precisely half of systems break → more interesting
- Lose if there's disagreement on your labels
  - you should provide clear, reasonable data

thanks to

Michael Hicks

Michelle Mazurek

Nick Diakopoulos

the rest of the original build it break it fix it team

questions? participants?

- Running pilot now with sentiment
- Talk to me, Sudha or Allyson if you want to join

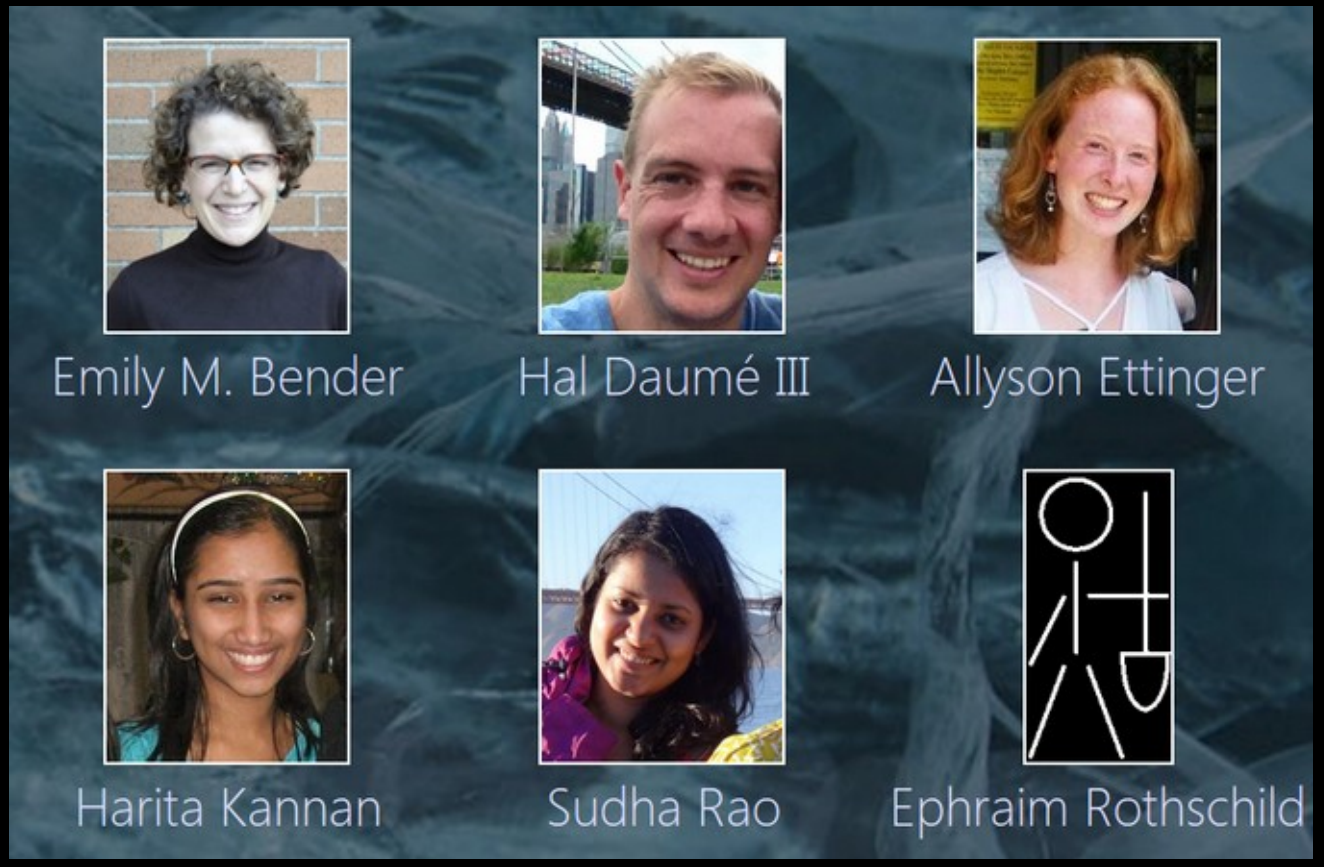

Photo credit:
Pavel Blazek

bibinlp.umiacs.umd.edu