Testing, Validation and Evaluation:

How Do You Know if Your NLP System Actually Works?

Dr. Rachael Tatman
Senior Developer Advocate, Rasa

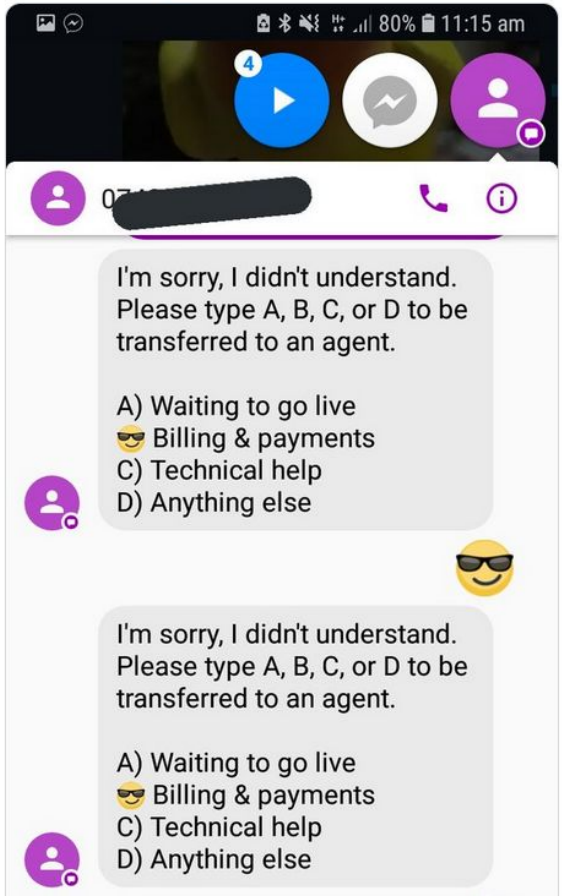
RASA

Hi, my name is Rachael and...

I work on chatbots

AroraXD
@AroraXD

I hate chatbots 🤖



Owen
@0w3nl

I Hate ChatBots so badly. My last 10 mins:
Fedex ChatBot: 1pm or 1am?
Me: 1pm
ChatBot: did you say 1am?
Me: 1pm
ChatBot: did you say 1am?
Me: 1pm
ChatBot: did you say 1am?
Me: 1pm
ChatBot: did you say 1am?
Me: 1pm
ChatBot: did you say 1am?
Me: 1pm
ChatBot: did you say 1am?
.....

6:19 AM · Aug 20, 2021 · Twitter Web App

Systems people hate using are a failure of engineering.

How can you build, deploy and maintain NLP systems that work?

1. Testing
2. Validation
3. Evaluation

Testing

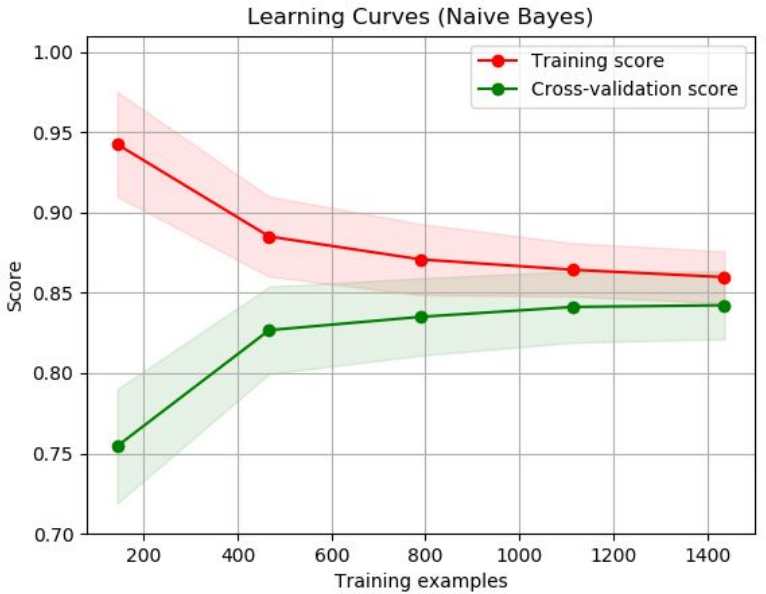
- For stable behaviour
- Unit and integration tests
- **Chatbot example**: fixed conversations that should always have the exact same outcome

tests/test_stories.yml

```
stories:
- story: A basic story test
  steps:
  - user: |
      hello
    intent: greet
  - action: utter_ask_howcanhelp
  - user: |
      show me [chinese]{"entity": "cuisine"} restaurants
    intent: inform
  - action: utter_ask_location
  - user: |
      in [Paris]{"entity": "location"}
    intent: inform
  - action: utter_ask_price
```

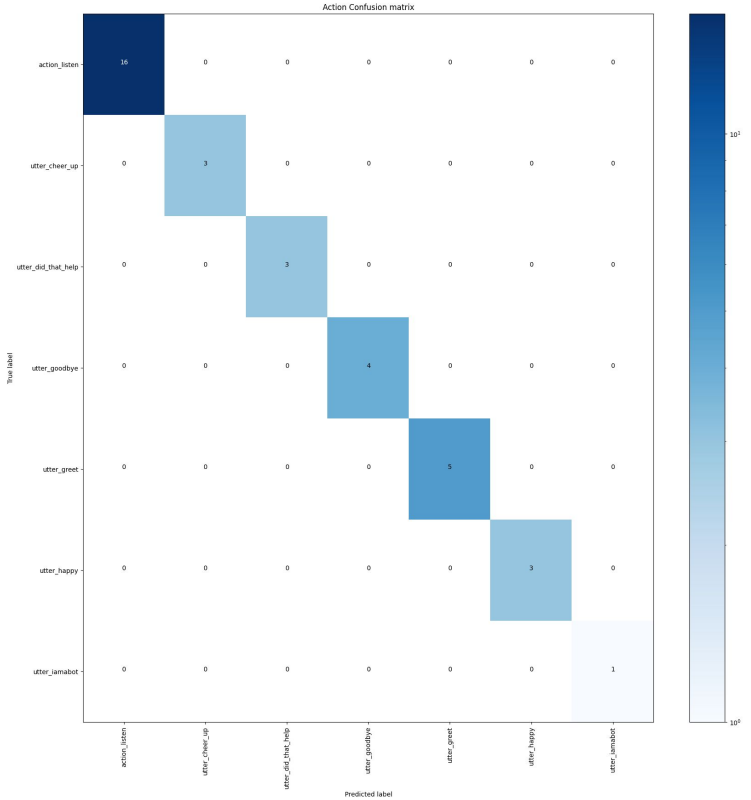
Validation

- Key to avoiding over- and underfitting
- Evaluates model performance when presented with novel data
- Train/test split is just the *first* stage of validation

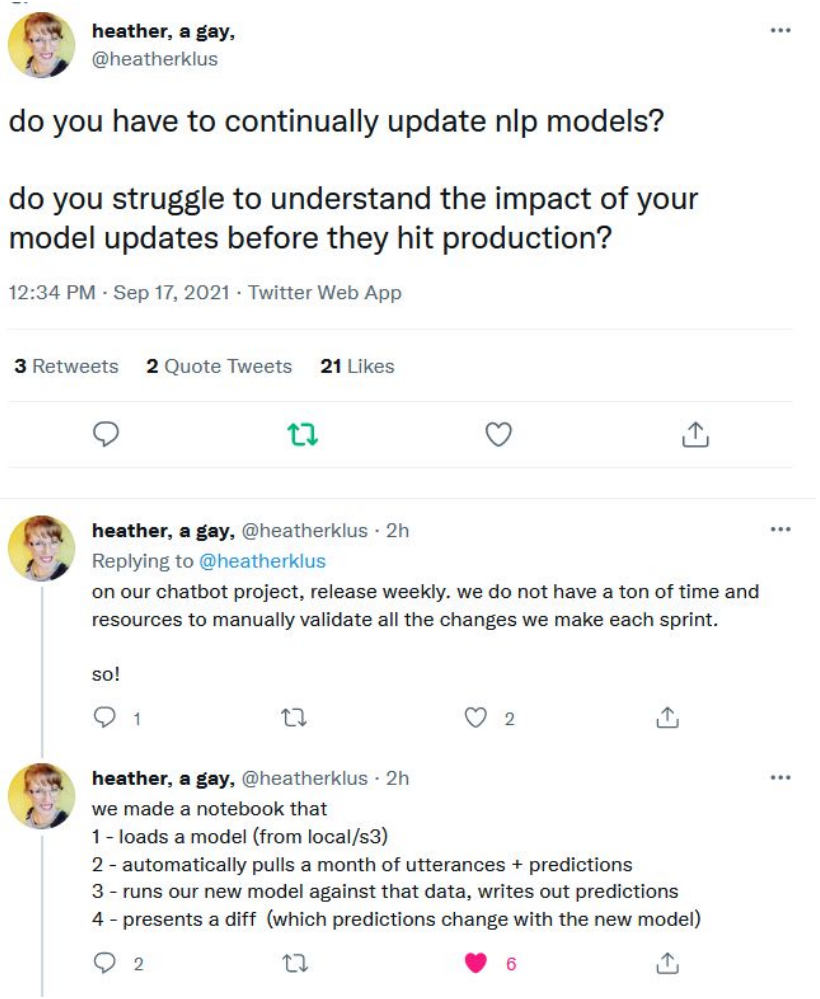


Validation

- In any application where you expect changes over time (hint; all systems interacting with language) you'll need to revalidate with timely data
- Validation data should *match the distribution of data your systems sees in production* (which, luckily, you have)



- Heather is a Principal Machine Learning Engineer @ T-Mobile
- Her team's assistant serves more than 2 million insights/day
- A third of users **choose to use the chatbot instead of talking to a person**



Evaluation

- You can have 100% test coverage & the world's best cross-validation... but if your system isn't effectively solving a problem then it doesn't matter
- **What does success look like?**
  - Your company saves money
  - User satisfaction is high
  - We save user's time (<- #1 factor that makes people actually use a chatbot*)
- This is extremely system dependent but should be the **first** thing you figure out when you scope your project

*Brandtzaeg, P. B., & Følstad, A. (2017, November). Why people use chatbots. In International conference on internet science (pp. 377-392). Springer, Cham.

RASA

OK, cool, that's all very theoretical, what does it actually look like in practice?

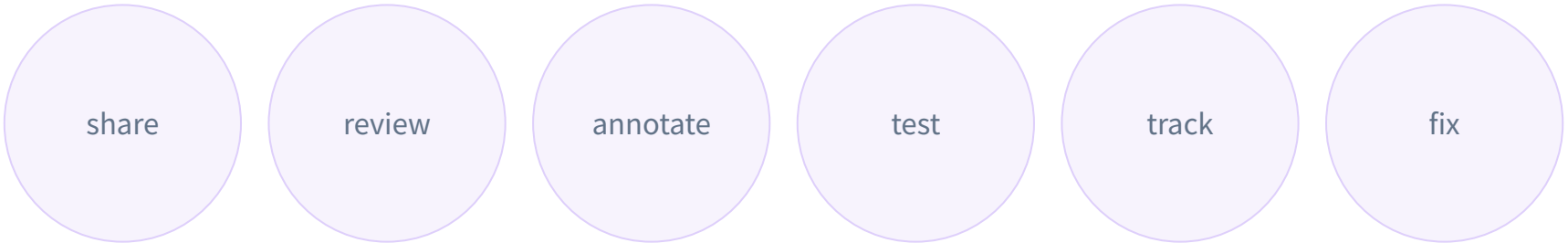
Conversation-Driven Development (CDD)

A chatbot example:
Conversation Driven Development

- Human-labeller-in-the-deployment-loop
- You have to look at your data
- CICD but including data labelling & retraining
- I'm not kidding: you **have** to look at your data

ABOUT CDD

Conversation-Driven Development is made up of six actions



RASA

Share

Users will always surprise you.

So get some test users to try your prototype **as early as possible**.

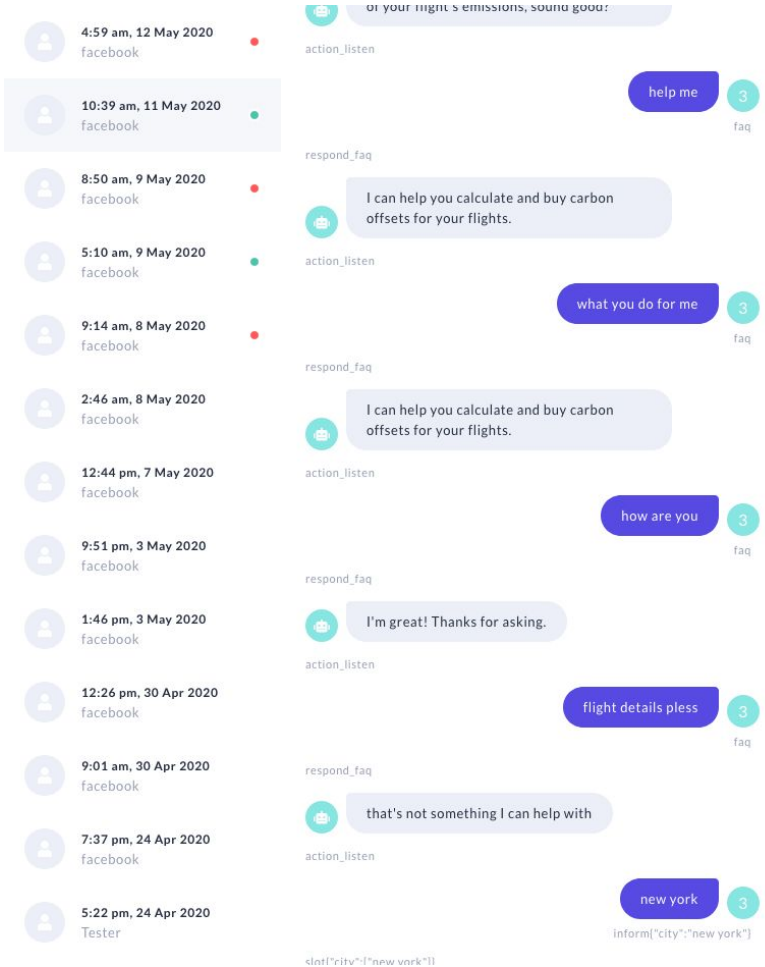
Shipping without having a bunch of test users has never worked. Your project won't be the exception.

RASA

Review

At every stage of a project, it is worth reading what users are saying.

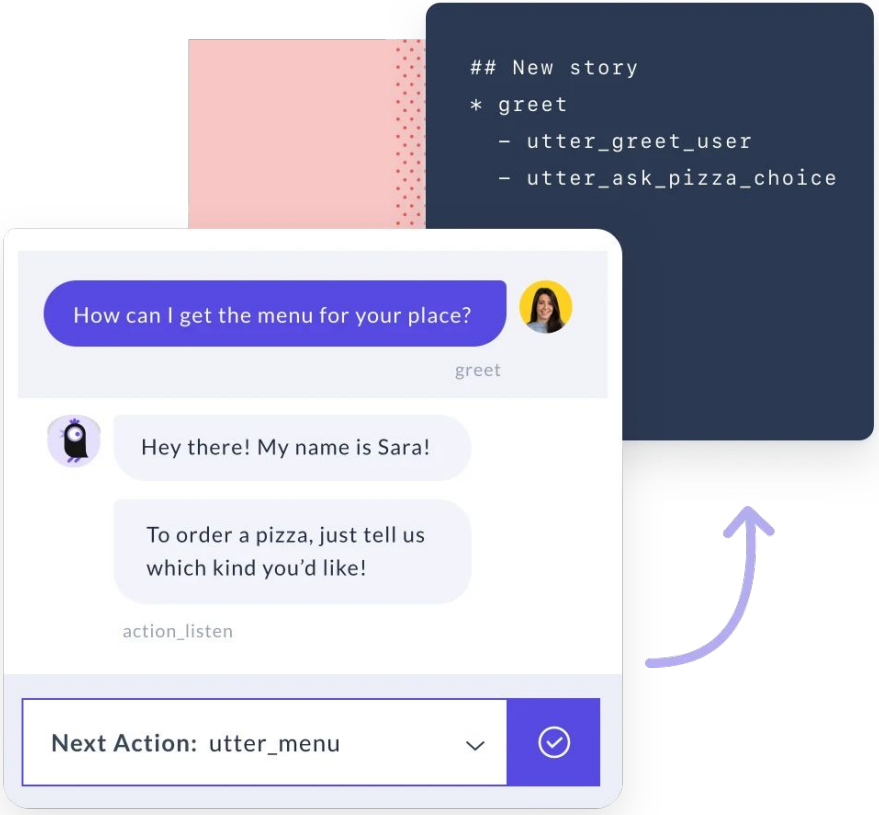
Avoid getting caught up in metrics right away. Conversations are valuable data.



Annotate

Using a script to generate synthetic training data

Turning real messages into training examples

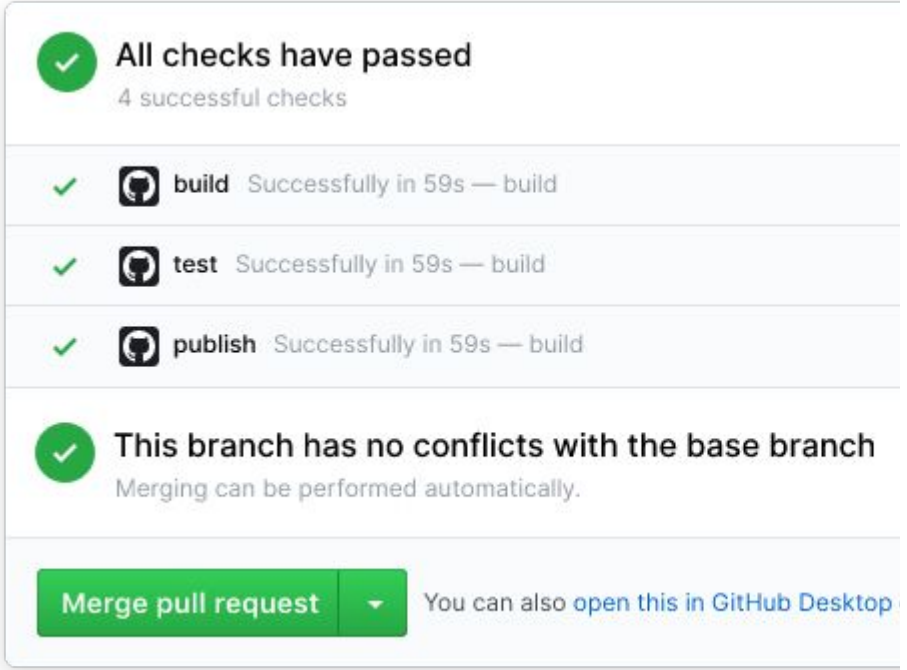


Test

Professional teams don’t ship applications without tests.

Use whole conversations as end-to-end tests

Run them on a continuous integration (CI) server.



Track

Use proxy measures to track which conversations are successful and which ones failed.

‘Negative’ signals are useful too, e.g. users **not** getting back in touch with support.

Fix

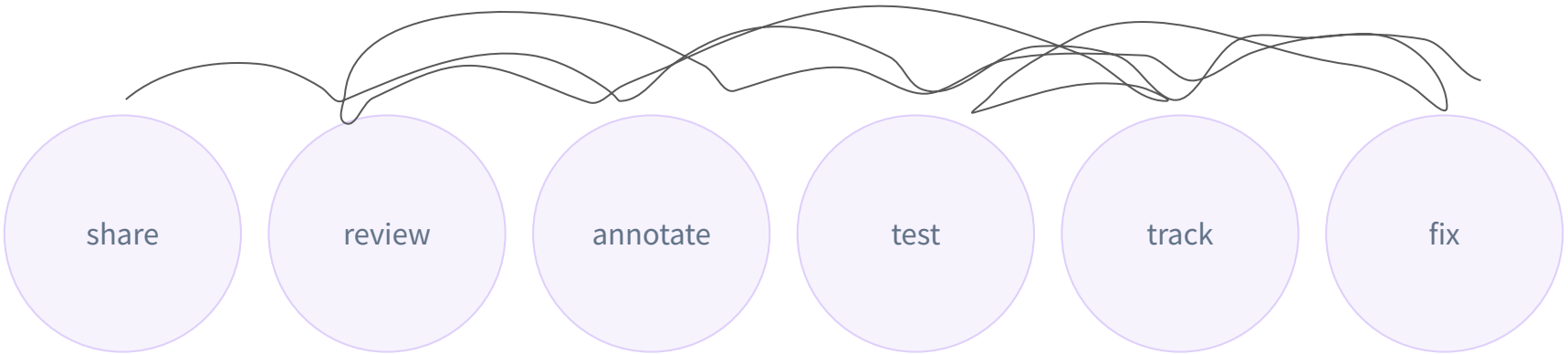
Study conversations that went smoothly and ones that failed.

Successful conversations can become new tests 🎉

Fix issues by annotating more data and/or fixing your code 🛠️

ABOUT CDD

Conversation-Driven Development is made up of six actions



You're going to have a lot of iteration & improvement, both before and after launch

1. Testing
2. Validation
3. Evaluation

Thanks! Questions?

@rctatman

RASA