

Testing, Validation and Evaluation:

**How Do You Know if Your NLP System
Actually Works?**

Dr. Rachael Tatman
Senior Developer Advocate, Rasa

**Hi, my name is Rachael
and...**

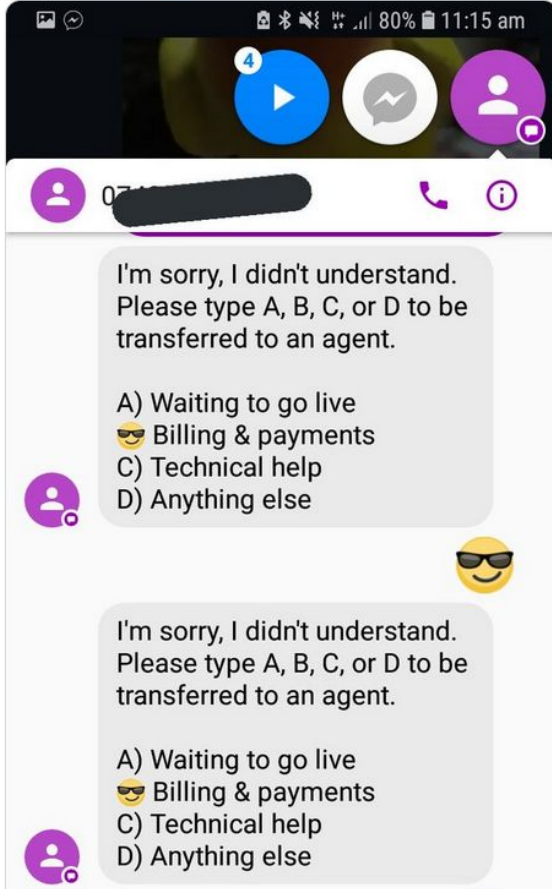
I work on chatbots



AroraXD
@AroraXD



I hate chatbots 🤖





Owen

@Ow3nl

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

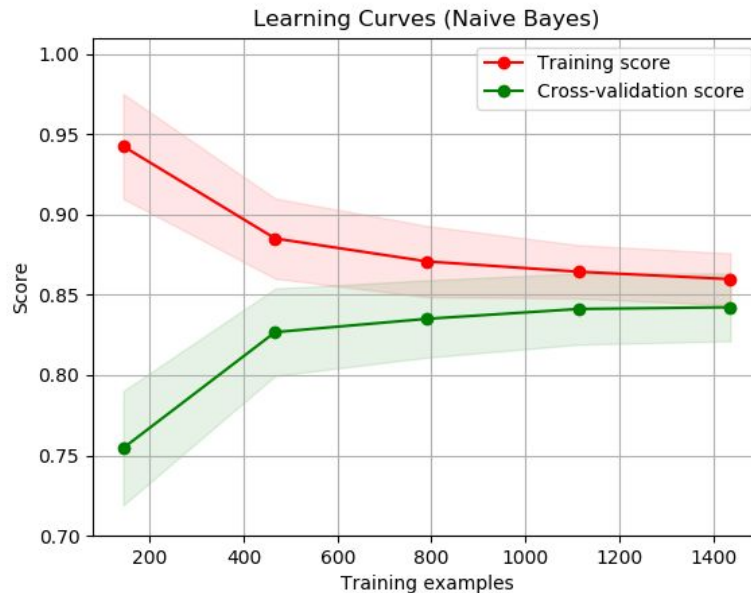
- For stable behaviour
- Unit and integration tests
- **Chatbot example:** fixed conversations that should always have the exact same outcome

```
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

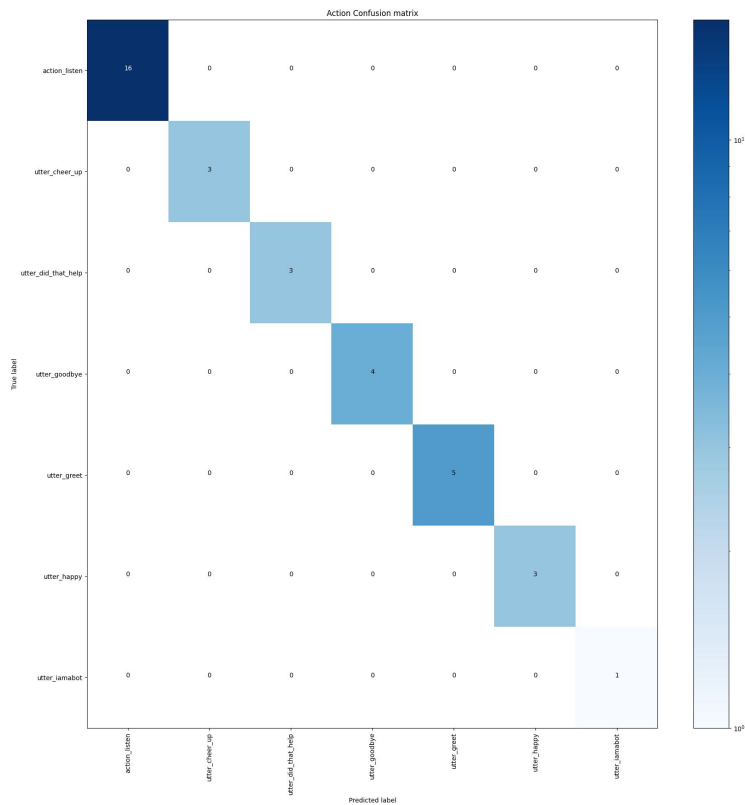
- Key to avoiding over- and underfitting
- Evaluates model performance when presented with novel data
- Train/test split is just the *first* stage of validation



scikit-learn developers, BSD <<http://opensource.org/licenses/bsd-license.php>>, via Wikimedia Commons

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**



heather, a gay,
@heatherklus

do you have to continually update nlp models?

do you struggle to understand the impact of your model updates before they hit production?

12:34 PM · Sep 17, 2021 · Twitter Web App

3 Retweets 2 Quote Tweets 21 Likes



heather, a gay, @heatherklus · 2h

Replying to @heatherklus

on our chatbot project, release weekly. we do not have a ton of time and resources to manually validate all the changes we make each sprint.

so!



1



2



heather, a gay, @heatherklus · 2h

we made a notebook that

1 - loads a model (from local/s3)

2 - automatically pulls a month of utterances + predictions

3 - runs our new model against that data, writes out predictions

4 - presents a diff (which predictions change with the new model)



2



6



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.

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

Conversation-Driven Development is made up of six actions



share

review

annotate

test

track

fix

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.



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.

4:59 am, 12 May 2020
facebook
action_listen

10:39 am, 11 May 2020
facebook
respond_faq

8:50 am, 9 May 2020
facebook
action_listen

5:10 am, 9 May 2020
facebook
action_listen

9:14 am, 8 May 2020
facebook
respond_faq

2:46 am, 8 May 2020
facebook
action_listen

12:44 pm, 7 May 2020
facebook
action_listen

9:51 pm, 3 May 2020
facebook
respond_faq

1:46 pm, 3 May 2020
facebook
action_listen

12:26 pm, 30 Apr 2020
facebook
respond_faq

9:01 am, 30 Apr 2020
facebook
respond_faq

7:37 pm, 24 Apr 2020
facebook
action_listen

5:22 pm, 24 Apr 2020
Tester
inform[{"city":"new york"}]

of your night's emissions, sound good?

help me

I can help you calculate and buy carbon offsets for your flights.

what you do for me

I can help you calculate and buy carbon offsets for your flights.

how are you

I'm great! Thanks for asking.

flight details pless

that's not something I can help with

new york

Annotate



Using a script to generate synthetic training data



Turning real messages into training examples

The image shows a chat interface and a code editor. The chat interface has a white background with a blue header. A user message (blue bubble) says "How can I get the menu for your place?" with a profile picture of a woman and the label "greet". A bot response (light blue bubble) says "Hey there! My name is Sara!" with a penguin profile picture. Below it, another bot response (light blue bubble) says "To order a pizza, just tell us which kind you'd like!" with the label "action_listen". At the bottom, a white box contains "Next Action: utter_menu" with a dropdown arrow and a blue button with a white checkmark. To the right, a dark blue code editor shows a script:

```
## New story
* greet
- utter_greet_user
- utter_ask_pizza_choice
```

 A red box with a dotted pattern is partially visible behind the code editor. A purple arrow points from the chat interface towards the code editor.

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.



All checks have passed

4 successful checks



build Successfully in 59s — build



test Successfully in 59s — build



publish Successfully in 59s — build



This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#)

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.



**Conversation between
Demobot and
1599192453538535**

link-1-clicked

form_failed_...



Fix



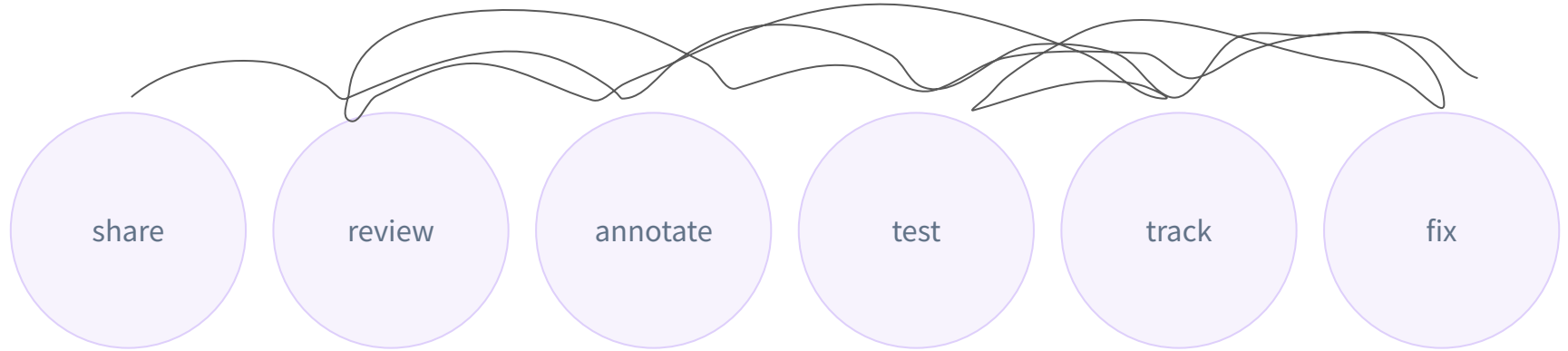
Study conversations that went smoothly and ones that failed.

Successful conversations can become new tests 🎉

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



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