PyData
New York City 2017

—

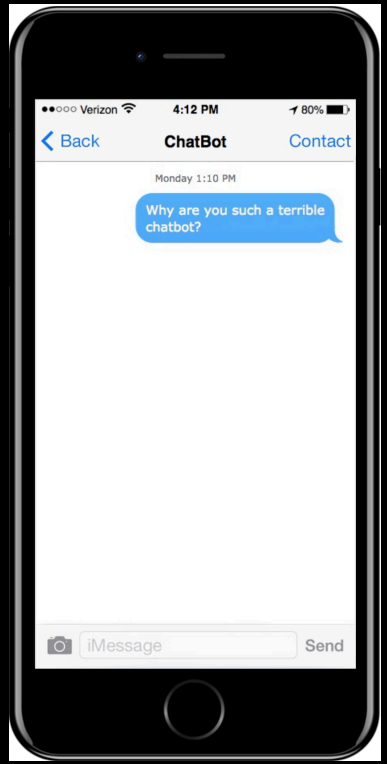
Nick Acosta
Developer Advocate

Nick Acosta

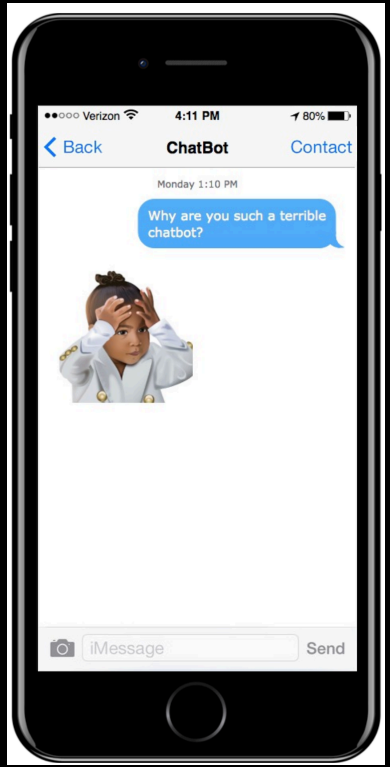
Before becoming a Developer Advocate at IBM, Nick studied computer science at Purdue University and the University of Southern California, and was a high performance computing consultant for Hewlett-Packard in Grenoble, France. He now specializes in machine learning and interacting with other data scientists of various communities, startups, and enterprises in order to help them succeed on IBM's data science platform. He has a strong interest in data science education and all things Kardashian.

Motivation for talk chatbots that respond with images

Motivation for talk chatbots that respond with images



Motivation for talk chatbots that respond with images



howdy.ai

Attachment upload API

Attachment upload API allows you to upload an attachment that you may later send out to many users, without having to repeatedly upload the same data each time it is sent :

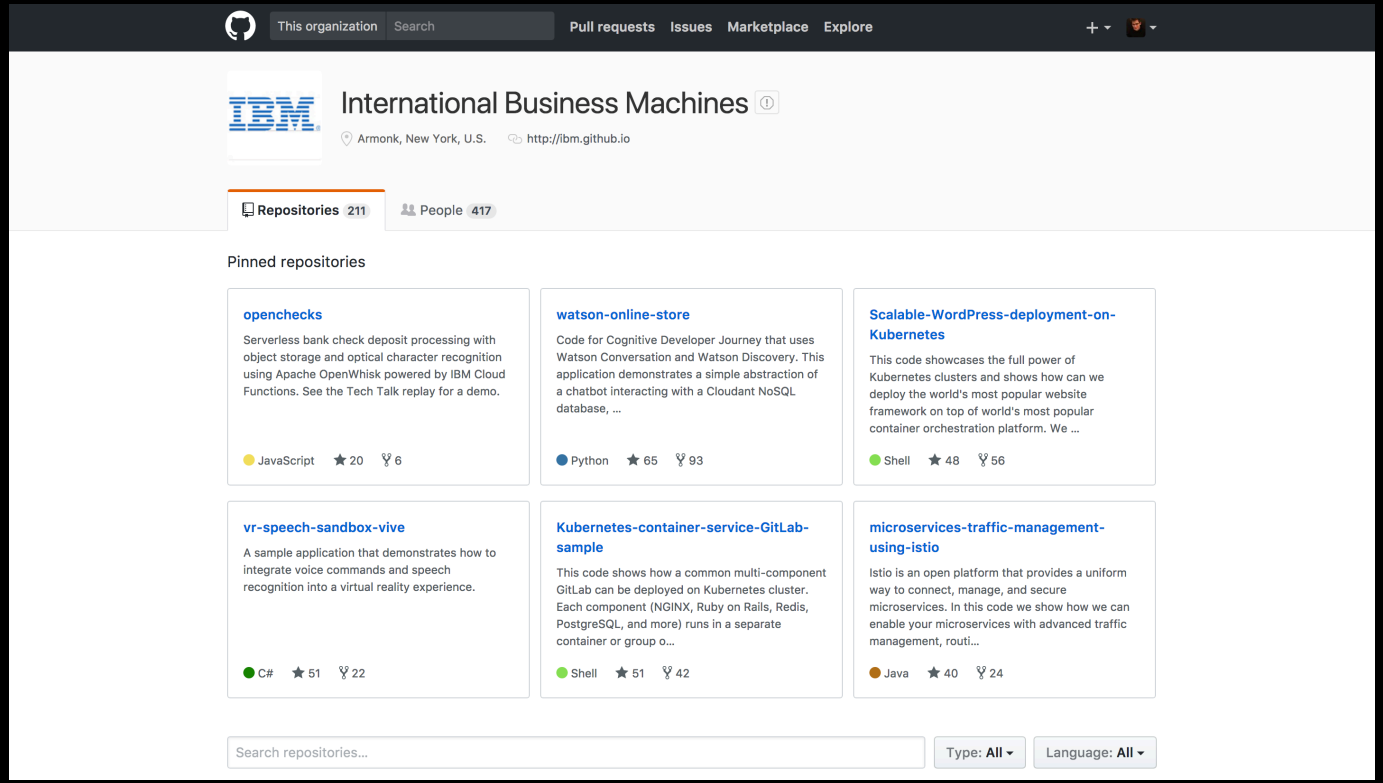
```
var attachment = {
        "type":"image",
        "payload":{
            "url":"https://pbs.twimg.com/profile_images/803642201653858305/IAW1DBPw_400x400.png",
            "is_reusable": true
        }
    };

    controller.api.attachment_upload.upload(attachment, function (err, attachmentId) {
        if(err) {
            // Error
        } else {
            var image = {
                "attachment":{
                    "type":"image",
                    "payload": {
                        "attachment_id": attachmentId
                    }
                }
            };
            bot.reply(message, image);
        }
    });
```

What am I even looking at?

Agenda

Part 2 The Data

IBM’s GitHub


This organization
Search
Pull requests
Issues
Marketplace
Explore
International Business Machines
Armonk, New York, U.S.
http://ibm.github.io
Repositories 211
People 417
Pinned repositories
openchecks
Serverless bank check deposit processing with object storage and optical character recognition using Apache OpenWhisk powered by IBM Cloud Functions. See the Tech Talk replay for a demo.
JavaScript 20 6
watson-online-store
Code for Cognitive Developer Journey that uses Watson Conversation and Watson Discovery. This application demonstrates a simple abstraction of a chatbot interacting with a Cloudant NoSQL database, ...
Python 65 93
Scalable-WordPress-deployment-on-Kubernetes
This code showcases the full power of Kubernetes clusters and shows how can we deploy the world's most popular website framework on top of world's most popular container orchestration platform. We ...
Shell 48 56
vr-speech-sandbox-vive
A sample application that demonstrates how to integrate voice commands and speech recognition into a virtual reality experience.
C# 51 22
Kubernetes-container-service-GitLab-sample
This code shows how a common multi-component GitLab can be deployed on Kubernetes cluster. Each component (NGINX, Ruby on Rails, Redis, PostgreSQL, and more) runs in a separate container or group o...
Shell 51 42
microservices-traffic-management-using-istio
Istio is an open platform that provides a uniform way to connect, manage, and secure microservices. In this code we show how we can enable your microservices with advanced traffic management, routi...
Java 40 24
Search repositories...
Type: All
Language: All

PyGitHub

```
from github import Github

# First create a Github instance:
g = Github("user", "password")

# Then play with your Github objects:
for repo in g.get_user().get_repos():
    print repo.name
```

plclassiferdata.ipynb

Part 3 The Models

Naïve Bayes Classifier

Bayes Theorem

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

Bayes Theorem

$$P(\text{programming language} = k|\text{text}) = \frac{P(\text{text}|\text{programming language} = k)P(\text{programming language} = k)}{P(\text{text})}$$

Bayes Theorem

$$P(\text{text}|\text{pl} = k) = ?$$

Naïve Assumption

$$P(\mathrm{word}_1, \mathrm{word}_2, \mathrm{word}_3, ... | \mathrm{pl} = k) = P(\mathrm{w}_1 | \mathrm{pl} = k)^{\#\mathrm{w}_1} P(\mathrm{w}_2 | \mathrm{pl} = k)^{\#\mathrm{w}_2} P(\mathrm{w}_3 | \mathrm{pl} = k)^{\#\mathrm{w}_3} ...$$

Naïve Assumption

$$P(\text{text}|\text{pl} = k) = ?$$

Naïve Assumption

$$P(\text{text}|\text{pl} = k) = ?$$

$$P(\text{text}|\text{pl} = k) = P(\text{word}_1, \text{word}_2, \text{word}_3, ...|\text{pl} = k)$$

Naïve Assumption

$$P(\text{text}|\text{pl} = k) = ?$$

$$P(\text{text}|\text{pl} = k) = P(\text{word}_1, \text{word}_2, \text{word}_3, ...|\text{pl} = k)$$

$$= P(\text{w}_1|\text{pl} = k)^{\#\text{w}_1} P(\text{w}_2|\text{pl} = k)^{\#\text{w}_2} P(\text{w}_3|\text{pl} = k)^{\#\text{w}_3} ...$$

Naïve Assumption

$$P(\text{text}|\text{pl} = k) = ?$$

$$P(\text{text}|\text{pl} = k) = P(\text{word}_1, \text{word}_2, \text{word}_3, ...|\text{pl} = k)$$

$$= P(\text{w}_1|\text{pl} = k)^{\#\text{w}_1} P(\text{w}_2|\text{pl} = k)^{\#\text{w}_2} P(\text{w}_3|\text{pl} = k)^{\#\text{w}_3} ...$$

$$= \Pi_i P(\text{w}_i|\text{pl} = k)^{\#\text{w}_i}$$

Naïve Bayes Classifier Prediction

$$P(\text{pl} = k|\text{text}) = \frac{P(\text{text}|\text{pl} = k)P(\text{pl} = k)}{P(\text{text})}$$

Naïve Bayes Classifier Prediction

$$P(\text{pl} = k|\text{text}) = \frac{P(\text{text}|\text{pl} = k)P(\text{pl} = k)}{P(\text{text})}$$

Can ignore P(text) as it does not change

$$P(\text{pl} = k|\text{text}) = P(\text{text}|\text{pl} = k)P(\text{pl} = k)$$

Naïve Bayes Classifier Prediction

$$P(\mathrm{pl} = k|\mathrm{text}) = \frac{P(\mathrm{text}|\mathrm{pl} = k)P(\mathrm{pl} = k)}{P(\mathrm{text})}$$

Can ignore P(text) as it does not change

$$P(\mathrm{pl} = k|\mathrm{text}) = P(\mathrm{text}|\mathrm{pl} = k)P(\mathrm{pl} = k)$$

$$= \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

Naïve Bayes Classifier Prediction

$$P(\mathrm{pl} = k|\mathrm{text}) = \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

Naïve Bayes Classifier Prediction

$$P(\mathrm{pl} = k|\mathrm{text}) = \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

$$\log P(\mathrm{pl} = k|\mathrm{text}) = \log \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

Naïve Bayes Classifier Prediction

$$P(\mathrm{pl} = k|\mathrm{text}) = \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

$$\log P(\mathrm{pl} = k|\mathrm{text}) = \log \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

$$= \sum_i \log \#w_i P(w_i|\mathrm{pl} = k) + \log P(\mathrm{pl} = k)$$

Naïve Bayes Classifier Prediction

$$P(\mathrm{pl} = k|\mathrm{text}) = \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

$$\log P(\mathrm{pl} = k|\mathrm{text}) = \log \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

$$= \sum_i \log \#w_i P(w_i|\mathrm{pl} = k) + \log P(\mathrm{pl} = k)$$

$$Y^* = \mathrm{argmax}_k \sum_i \log \#w_i P(w_i|\mathrm{pl} = k) + \log P(\mathrm{pl} = k)$$

Naïve Bayes Classifier Training

$$P(\mathrm{pl} = k|\mathrm{text}) = \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

Naïve Bayes Classifier Training

$$P(\mathrm{pl} = k|\mathrm{text}) = \Pi_i P(\mathrm{w}_i|\mathrm{pl} = k)^{\#\mathrm{w}_i} P(\mathrm{pl} = k)$$

For each programming language:
- Probability it occurs in data set (1)
- Probability of every word given its in programming language k (2)

1 $$\frac{\text{count of programs of lang k}}{\text{count of programs}}$$

2 $$\frac{\text{occurrences of word i in pl k}}{\text{words in pl k}}$$

Watson Natural Language Classifier

Watson Natural Language Classifier

- Training
  - 1 - Call an API
- Testing
  - 1 - Call an API

plclassifiermodels.ipynb

Part 4 The End

New Developments in Programming Language Classification

GitHub's own Programming Language Classifier, Linguist

Linguist

This library is used on GitHub.com to detect blob languages, ignore binary or vendored files, suppress generated files in diffs, and generate language breakdown graphs.

See Troubleshooting and `CONTRIBUTING.md` before filing an issue or creating a pull request.

Troubleshooting

My repository is detected as the wrong language

The Language stats bar displays languages percentages for the files in the repository. The percentages are calculated based on the bytes of code for each language as reported by the List Languages API. If the bar is reporting a language that you don't expect:

1. Click on the name of the language in the stats bar to see a list of the files that are identified as that language.
2. If you see files that you didn't write, consider moving the files into one of the paths for vendored code, or use the manual overrides feature to ignore them.
3. If the files are being misclassified, search for open issues to see if anyone else has already reported the issue. Any information you can add, especially links to public repositories, is helpful.
4. If there are no reported issues of this misclassification, open an issue and include a link to the repository or a sample of the code that is being misclassified.

Microsoft's Text Classification Kaggle Competition for Malware Classification

Moving away from DSL in Program Synthesis Research

Published as a conference paper at ICLR 2017

DEEPCODER: LEARNING TO WRITE PROGRAMS

Matej Balog*
Department of Engineering
University of Cambridge

Alexander L. Gaunt, Marc Brockschmidt,
Sebastian Nowozin, Daniel Tarlow
Microsoft Research

ABSTRACT

We develop a first line of attack for solving programming competition-style problems from input-output examples using deep learning. The approach is to train a neural network to predict properties of the program that generated the outputs from the inputs. We use the neural network's predictions to augment search techniques from the programming languages community, including enumerative search and an SMT-based solver. Empirically, we show that our approach leads to an order of magnitude speedup over the strong non-augmented baselines and a Recurrent Neural Network approach, and that we are able to solve problems of difficulty comparable to the simplest problems on programming competition websites.

1 INTRODUCTION

A dream of artificial intelligence is to build systems that can write computer programs. Recently, there has been much interest in program-like neural network models (Graves et al., 2014; Weston et al., 2015; Kurach et al., 2015; Joulin & Mikolov, 2015; Grefenstette et al., 2015; Sukhbaatar et al., 2015; Neelakantan et al., 2016; Kaiser & Sutskever, 2016; Reed & de Freitas, 2016; Zaremba et al., 2016; Graves et al., 2016), but none of these can *write programs*; that is, they do not generate human-readable source code. Only very recently, Riedel et al. (2016); Bunel et al. (2016); Gaunt et al. (2016) explored the use of gradient descent to induce source code from input-output examples via differentiable interpreters, and Ling et al. (2016) explored the generation of source code from unstructured text descriptions. However, Gaunt et al. (2016) showed that differentiable interpreter-based program induction is inferior to discrete search-based techniques used by the programming languages community. We are then left with the question of how to make progress on program induction using machine learning techniques.

Next Steps

Last thing

Attachment upload API

Attachment upload API allows you to upload an attachment that you may later send out to many users, without having to repeatedly upload the same data each time it is sent :

```
var attachment = {
        "type":"image",
        "payload":{
            "url":"https://pbs.twimg.com/profile_images/803642201653858305/IAW1DBPw_400x400.png",
            "is_reusable": true
        }
    };

    controller.api.attachment_upload.upload(attachment, function (err, attachmentId) {
        if(err) {
            // Error
        } else {
            var image = {
                "attachment":{
                    "type":"image",
                    "payload": {
                        "attachment_id": attachmentId
                    }
                }
            };
            bot.reply(message, image);
        }
    });
```

Q&A

Thank you

Nick Acosta
Developer Advocate
—
nacosta@us.ibm.com
github.com/PubChimps