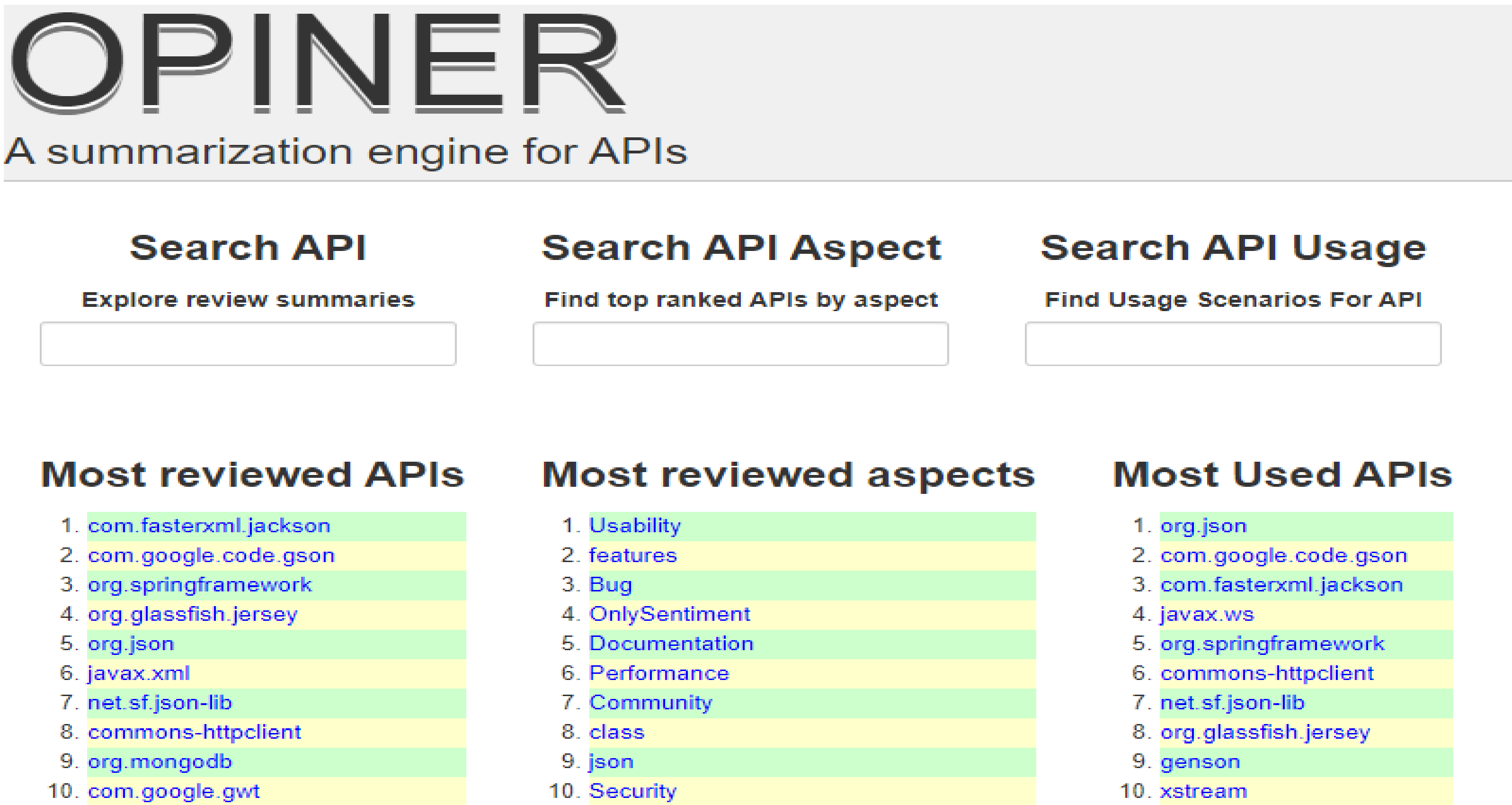
ENSF 612: Lecture 5
Course Project Overview

Dr. Gias Uddin, Assistant Professor
Department of Electrical and Computer Engineering,
University of Calgary.
https://giasuddin.ca/

Develop Supervised ML Models to Automatically Classify and Summarize Big Software Engineering Data



http://opiner.polymtl.ca/

Sources of big data – Software

Stack Exchange Online Technical Forums

- Across all of Stack Overflow and the Stack Exchange network, we saw **9+ billion pageviews** from **100+ million users** over the course of the year.

company JANUARY 18, 2019

State of the Stack 2019: A Year in Review

A loooong time ago, we used to post an annual "State of the Stack" update on the company and community. Then at some point it became an infographic which was… listen, everyone was doing infographics in 2011. Now it's 2019, the company has grown and changed in so many ways, so we're bringing this tradition…

David Fullerton
President and Chief Technology Officer (former)

StackExchange

All Sites Top Users Digests

All Technology Culture / Recreation Life / Arts Science Professional Business

Sort by: Traffic

Stack Overflow Q&A for professional and enthusiast programmers	20m questions	30m answers	70% answered	13m users	9.8m visits/day	6.7k questions/day	12y2m site age
Super User Q&A for computer enthusiasts and power users	449k questions twitter	646k answers	66% answered	932k users	654k visits/day	119 questions/day	11y2m site age

Sources of big data - Software

100+ millions GitHub software repositories

40m+

developers on GitHub, including 10M new users in 2019.*

44m+

repositories created in the last year—and 44% more developers created their first repository in 2019 than in 2018.*

87m+

pull requests merged in the last year—and 28% more developers opened their first pull request in 2019 than in 2018.*

20m+

issues closed in the last year. That's a lot of decisions made, bugs fixed, and boxes checked.*

https://octoverse.github.com/

Project Steps Per Group

1. Pick a research paper that labeled/summarized software engineering dataset
2. Collect their dataset that they shared and that they used for labeling the dataset
3. Add 1000 more records into the dataset (details later)
4. Replicate the developed ML models in the paper using the original + extended dataset.
5. Extend the papers by experimenting with additional ML models
6. Show how the findings from ML models can be used to summarize the datasets and similar data in software engineering, e.g., by following Opiner

Example Project Idea 1

Automating Intention Mining

Qiao Huang, Xin Xia, David Lo, Gail C. Murphy

Abstract—Developers frequently discuss aspects of the systems they are developing online. The comments they post to discussions form a rich information source about the system. Intention mining, a process introduced by Di Sorbo et al., classifies sentences in developer discussions to enable further analysis. As one example of use, intention mining has been used to help build various recommenders for software developers. The technique introduced by Di Sorbo et al. to categorize sentences is based on linguistic patterns derived from two projects. The limited number of data sources used in this earlier work introduces questions about the comprehensiveness of intention categories and whether the linguistic patterns used to identify the categories are generalizable to developer discussion recorded in other kinds of software artifacts (e.g., issue reports).
To assess the comprehensiveness of the previously identified intention categories and the generalizability of the linguistic patterns for category identification, we manually created a new dataset, categorizing 5,408 sentences from issue reports of four projects in GitHub. Based on this manual effort, we refined the previous categories. We assess Di Sorbo et al.'s patterns on this dataset, finding that the accuracy rate achieved is low (0.31). To address the deficiencies of Di Sorbo et al.'s patterns, we propose and investigate a convolution neural network (CNN)-based approach to automatically classify sentences into different categories of intentions. Our approach optimizes CNN by integrating batch normalization to accelerate the training speed, and an automatic hyperparameter tuning approach to tune appropriate hyperparameters of CNN. Our approach achieves an accuracy of 0.84 on the new dataset, improving Di Sorbo et al.'s approach by 171%. We also apply our approach to improve an automated software engineering task, in which we use our proposed approach to rectify misclassified issue reports, thus reducing the bias introduced by such data to other studies. A case study on four open source projects with 2,076 issue reports shows that our approach achieves an average AUC score of 0.687, which improves other baselines by at least 16%.

Example Project Idea 2

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. X, NO. X, NOVEMBER 2018 1

Automatic Mining of Opinions Expressed About APIs in Stack Overflow

Gias Uddin and Foutse Khomh

Abstract—With the proliferation of online developer forums, developers share their opinions about the APIs they use. The plethora of such information can present challenges to the developers to get quick but informed insights about the APIs. To understand the potential benefits of such API reviews, we conducted a case study of opinions in Stack Overflow using a benchmark dataset of 4522 sentences. We observed that opinions about diverse API aspects (e.g., usability) are prevalent and offer insights that can shape developers' perception and decisions related to software development. Motivated by the finding, we built a suite of techniques to automatically mine and categorize opinions about APIs from forum posts. First, we detect opinionated sentences in the forum posts. Second, we associate the opinionated sentences to the API mentions. Third, we detect API aspects (e.g., performance, usability) in the reviews. We developed and deployed a tool called Opiner, supporting the above techniques. Opiner is available online as a search engine, where developers can search for APIs by their names to see all the aggregated opinions about the APIs that are automatically mined and summarized from developer forums.

Example Project Idea 3

Categorizing the Content of GitHub README Files

Gede Artha Azriadi Prana · Christoph Treude · Ferdian Thung · Thushari Atapattu · David Lo

Received: date / Accepted: date

Abstract README files play an essential role in shaping a developer's first impression of a software repository and in documenting the software project that the repository hosts. Yet, we lack a systematic understanding of the content of a typical README file as well as tools that can process these files automatically. To close this gap, we conduct a qualitative study involving the manual annotation of 4,226 README file sections from 393 randomly sampled GitHub repositories and we design and evaluate a classifier and a set of features that can categorize these sections automatically. We find that information discussing the 'What' and 'How' of a repository is very common,

Project Steps Per Group – Idea

1. Additional project ideas will be appreciated. Teams with new project ideas will be given higher preference
2. If more than one team picks same project idea, the teams need to differ from each other on the following key aspects:
   - The additional data to label (each team should label data different from each other)
   - The ML models that will be experimented with
3. If two teams are using the same project idea, the team with higher data quality and higher ML model performance will be given more grade than the other team.

Project Steps Per Group – Data Labeling Instructions

1. Each team will add at least 1000 new records to an existing dataset.
2. Each team will label the new 1000 records following approaches similar to the paper as described in the paper
3. Each team member will label the 1000 records separately as follows:
    1. Each member will label each of 1000 records
    2. For each record, the final label would be the one that most agree (e.g., two say l1 and one say l2, then the label is l1)
    3. For each record, if no such majority is found, a final label will be picked based on further discussions among the members