

# Tellina: A Deep Learning Powered End User Scripting Assistant

Keynote - NLC2CMD Challenge @NeurIPS'20



Victoria Lin

Salesforce AI Research

Work done at Paul G. Allen School of Computer Science & Engineering

Natural Language → Programming Langauge



# http://tellina.rocks

TELLINA

Type a natural language command...

Sample Questions   About

Recently Asked Questions

- delete all files**  
⌚ 2020-12-11 21:34:50 📍 São Paulo, BR  

```
find . -delete
```

[find](#) [x](#)
- clear terminal**  
⌚ 2020-12-11 21:33:44 📍 São Paulo, BR  

```
clear
```

[clear](#) [x](#)
- print a to the screen**  
⌚ 2020-12-11 21:31:30 📍 São Paulo, BR  

```
echo -n $( cat )
```

[cat](#) [echo](#) [x](#)
- find files in current dir**  
⌚ 2020-12-11 21:30:55 📍 São Paulo, BR  

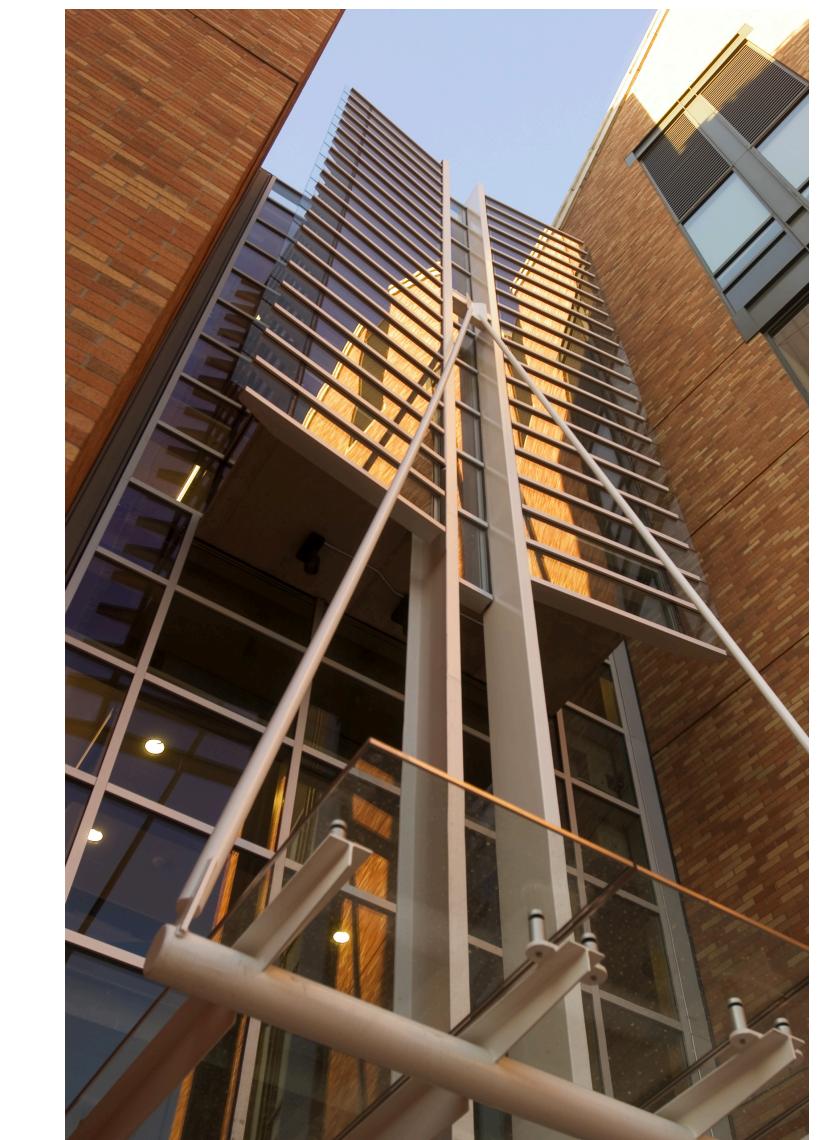
```
find . | xargs -I {} grep '^use strict' {}
```

[grep](#) [xargs](#) [find](#) [x](#)

**Tips**

- Express your goal in English and get its Bash translation. For example,
  - "split '/usr/bin/gcc' into 1000 files of about equal size";
  - "print top 10 largest files and directories";
  - [more examples...](#)
- Tellina works best when the input is specific. For example,
  - instead of "some files", use "pdf files";
  - instead of "sometime ago", use "5 days ago".
- Quoting `'''` names, strings and regular expressions can improve the translation.
- Put `/` at the end of a directory name such that Tellina interprets it as a folder.

© Tellina Tools 2018-2020. If you have any suggestions about the project, [email us](#) or issue a pull request on [Github](#).



Paul G. Allen Center of  
Computer Science &  
Engineering. 2016.01-2017.09

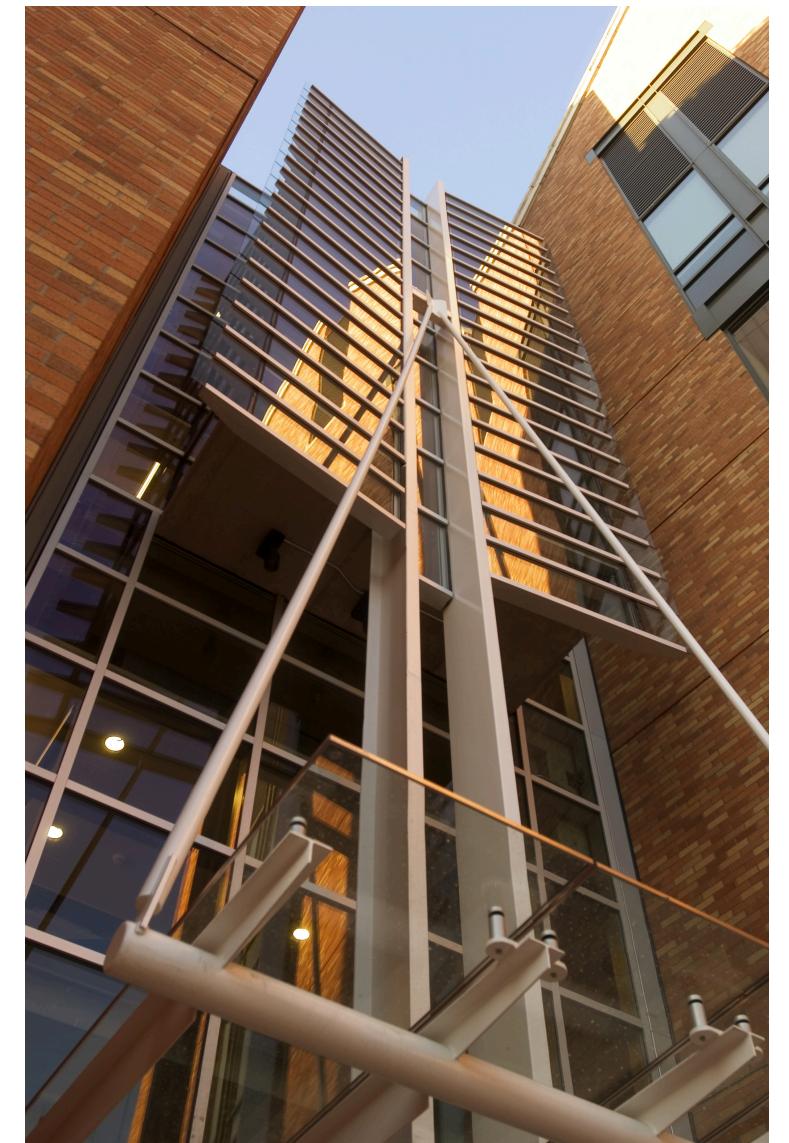
# http://tellina.rocks

```
find "myDir/" -mtime -24h | xargs -n [number] -I {} ls -l {}
```

```
find "myDir/" -mtime -24h -print0 | xargs -0 -n [number] -I {} ls -l {}
```

```
find "myDir/" -mtime -24h | xargs -I {} ls -l -d {}
```

```
find "myDir/" -mtime -24h | xargs -r -I {} ls -l -d {}
```



Paul G. Allen Center of  
Computer Science &  
Engineering. 2016.01-2017.09

# Inspiration



write down a command-line to see the help text that matches each argument  
try [showthedocs](#) for explaining other languages

 EXPLAIN

## examples

- :(){ :|:& };:
- for user in \$(cut -f1 -d: /etc/passwd); do crontab -u \$user -l 2>/dev/null; done
- file=\$(echo `basename "\$file"`)
- true && { echo success; } || { echo failed; }
- cut -d ' ' -f 1 /var/log/apache2/access\_logs | uniq -c | sort -n
- tar zcf - some-dir | ssh some-server "cd /; tar xvzf -"
- tar xzvf archive.tar.gz
- find . -type f -print0
- ssh -i keyfile -f -N -L 1234:www.google.com:80 host
- git log --graph --abbrev-commit --pretty=oneline origin..mybranch



# Inspiration

**explainshell.com**

about  theme 

write down a command-line to see the help text that matches each argument  
try [showthedocs](#) for ex...

**COMMANDLINEFU.COM** grep the archive...

+New Commands Login

All commands (14,037) sorted by date

**Individually 7zip all files in current directory** 0

```
$ for i in *.*; do 7z a "$i".7z "$i"; done
```

Romero · 2020-12-07 10:02:42

**Moving large number of files** 0

```
$ find /source/directory -mindepth 1 -maxdepth 1 -name '*' -print0 | xargs -0 mv -t /target/directory;
```

aysadk · 2020-11-17 12:30:45

**Show contents of all git objects in a git repo** 0

```
$ find .git/objects/ -type f \| sed 's/\.\git\.\objects\.\///' | sed 's/\///g' | xargs -n1 -I% echo echo "%" \$\(\git cat-file -p "%"\) \0 | xargs -n1 -0 sh -c
```

wires · 2020-11-15 09:24:58

**Rename all subtitles files with the same name of mp4 files in same folder** 1

```
$ paste -d < (ls -1 *.mp4) < (ls -1 *.srt) | while read line; do movie="\${line%%.*}"; subtitle="\${line##*.}"; mv "\${subtitle}" "\${movie%.*}.srt"; done
```

ricardofunke · 2020-11-08 02:47:13

**tree command limit depth for recursive directory list** 2

```
$ tree -L 2 -u -g -p -d
```

aysadk · 2020-11-04 10:45:39

Assume that you have a form , in the so...  
Convert a string to  
Write comments to your history.  
A null operation with the name 'commen...'  
Diff remote webpages using wget  
Create a persistent remote Proxy serve...  
Create a persistent remote Proxy server ...  
Using ASCII Art output on MPlayer  
Not so useful. Just a cool feature.  
tar directory and compress it with show...  
tar directory and compress it with showi...  
Which processes are listening on a spe...  
swap out "80" for your port of interest. C...  
Display error pages in report format  
This command will return a full list of Er...  
SSH tunneling self-connection  
- port 8080 on localhost will be a SOCKS...

Stay in the loop...  
Follow the Tweets.  
 Every new command is wrapped in a tweet and posted to Twitter. Following the stream is a great way of staying abreast of the latest commands. For the more discerning, there are Twitter accounts for commands that get a minimum of 3 and 10 votes - that way only the great commands get tweeted.  
» <http://twitter.com/commandlinefu>  
» <http://twitter.com/commandlinefu3>  
» <http://twitter.com/commandlinefu10>

Subscribe to the feeds.  
 Use your favourite RSS aggregator to stay in touch with the latest commands. There are feeds mirroring the 3 Twitter streams as well as for virtually every other subset (users, tags, functions,...):  
Subscribe to the feed for:



# Inspiration

**explainshell.com**

about theme

write down a command-line to see the help text that matches each argument  
try [showthedocs](#) for ex

**COMMANDLINEFU.COM** grep the archive...

All commands (14,037) sorted by all time

**Individually 7zip all files in current directory**

```
$ for i in *.*; do 7z a "$i".7z "$i"; done
```

Romero · 2020-12-07 10:02:42

**Moving large number of files**

if you want to move with command mv large list of files than you would get following error /bin/mv:  
Argument list too long alternativite with exec: find /source/directory -mindepth 1 -maxdepth 1 -name '\*' -exec mv {} /target/directory \; Show Sample Output

```
$ find /source/directory -mindepth 1 -maxdepth 1 -name '*' -print0 | xargs -0 mv -t /target/directory;
```

aysadk · 2020-11-17 12:30:45

**Show contents of all git objects in a git repo**

This script finds all git objects and 'git cat-file's their content. This is really just a helper function to play around with the internals of git repositories. See <https://git-scm.com/book/en/v2/Git-Internals-Git-Objects> for more info. Show Sample Output

```
$ find .git/objects/ -type f \| sed 's/\.\git\.\objects\.\///' | sed 's/\.\//g' | xarg I% echo echo "%\$\\(git cat-file -p "%\$"\) \\0 | xargs -n1 -0 sh -c
```

wires · 2020-11-15 09:24:58

**Rename all subtitles files with the same name of mp4 files in same folder**

Use this command if you want to rename all subtitles for them to have the same name as the mp4 files.  
NOTE: The order of "ls -1 \*.mp4" must match the order of "ls -1 \*.srt", run the command bellow to make sure the \*.srt files will really match the movies after run this command: paste -d

```
$ paste -d <(ls -1 *.mp4) <(ls -1 *.srt) | while read line; do movie="${line%.*}"; subtitle="${line##*.}"; mv "${subtitle}" "${movie%.*}.srt"; done
```

ricardofunke · 2020-11-08 02:47:13

**tree command limit depth for recursive directory list**

sometimes I need list from path with max limit for recursive depth directory listing

```
$ tree -L 2 -u -g -p -d
```

aysadk · 2020-11-04 10:45:39

stackoverflow About Products For Teams Search... Log in Sign up

Home PUBLIC Stack Overflow Tags Users FIND A JOB Jobs Companies TEAMS What's this? Free 30 Day Trial

**Linux -How to delete all files in a directory without using find [closed]**

Asked 7 years, 6 months ago Active 2 years, 5 months ago Viewed 15k times

Closed. This question is off-topic. It is not currently accepting answers.

Want to improve this question? [Update the question](#) so it's on-topic for Stack Overflow.

Closed 7 years ago.

[Improve this question](#)

I am trying to delete all files in my directory "XYZ" without using find command in bash on Linux.

linux bash share follow edited Jul 4 '18 at 11:55 asked May 22 '13 at 13:59 Claudio tom 9,089 3 26 66

What about rm /your/dir/\*? It will delete all files (no directories). With rm -r /your/dir you will delete everything. – fedorqui 'SO stop harming' May 22 '13 at 14:01

1 when you asked a question, describe it clearly please. 1. if you want to only rm files, (excluding dir? link?) 2. do you want to do deletion recursively? – Kent May 22 '13 at 14:03

1 be very careful when you use rm -rf \* or any variation of it. – Bill May 22 '13 at 14:09

[add a comment](#)

**3 Answers**

» <http://twitter.com/commandlinefu>  
» <http://twitter.com/commandlinefu3>  
» <http://twitter.com/commandlinefu10>

**Subscribe to the feeds.**

Use your favourite RSS aggregator to stay in touch with the latest commands. There are feeds mirroring the 3 Twitter streams as well as for virtually every other subset (users, tags, functions,...):

Subscribe to the feed for:

The Overflow Blog

The semantic future of the web  
Podcast 294: Cleaning up build systems and gathering computer history

Featured on Meta

New Feature: Table Support

Unleash the power of IBM Cloud® with Intel® technologies.

See all servers → Report this ad

Jobs near you

Full Stack Software Engineer kaliber · kaliber.ai San Francisco, CA python javascript

Software Engineer, Full Stack Forward · San Francisco CA

TELLINA

**Data Driven**

Natural Language → Programming Langauge

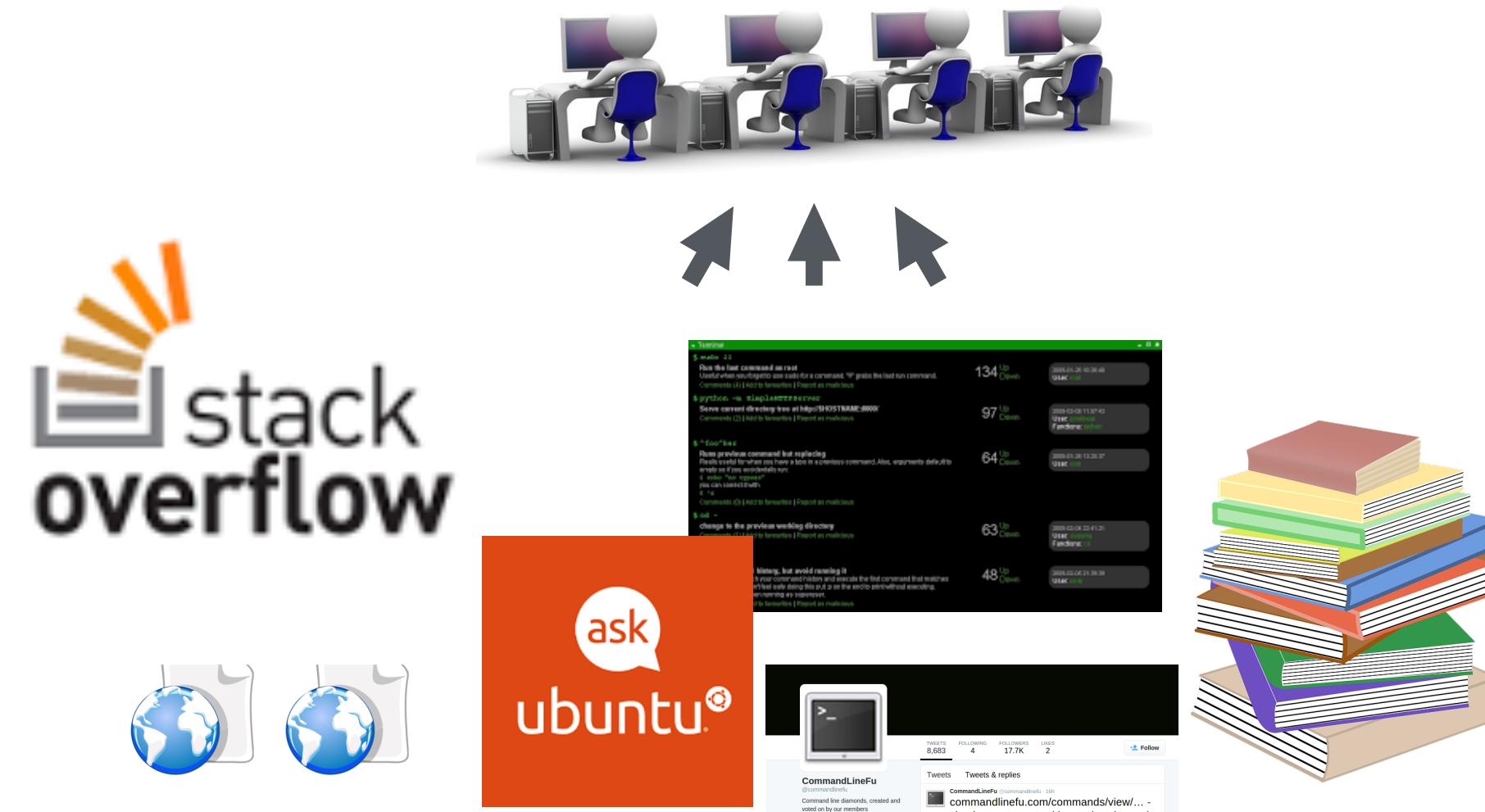


# Data Collection

NL2Bash is one of the ML tasks where expert knowledge is necessary to data collection.

**Experts:** Bash programmers hired from **Upwork™**

**Task:** Collect Bash commands from the web; (1) if a natural language description is found to accompany the Bash command, copy the NL description and modify it when appropriate; (2) otherwise, write an NL description for the Bash command.

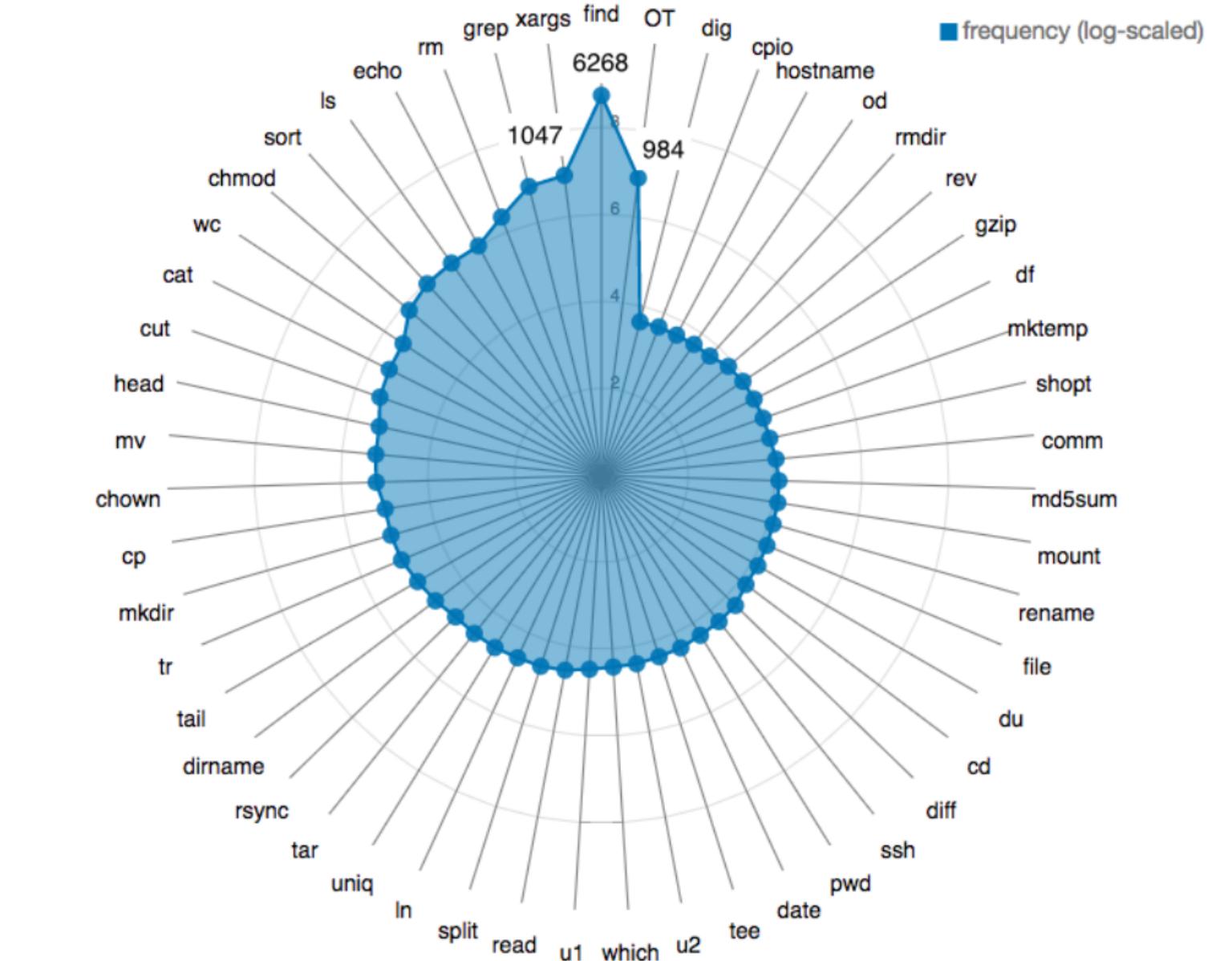


 Expert annotation produces high-quality data and is often necessary for authentic evaluation, but is difficult to scale.

# NL2Bash

By far the largest (~10k pairs) expert-annotated collection of Bash one-liners paired with their natural language descriptions.

Natural Language	Bash Command(s)
<i>find java files in the current directory tree that contain the pattern 'TODO' and print their names</i>	grep -l "TODO" *.java find . -name "*.java" -exec grep -il "TODO" {} \; find . -name "*.java"   xargs -I {} grep -l "TODO" {}
<i>display the 5 largest files in the current directory and its sub-directories</i>	find . -type f   sort -nk 5,5   tail -5 du -a .   sort -rh   head -n5 find . -type f -printf '%s %p\n'   sort -rn   head -n5
<i>search for all jpg images on the system and archive them to tar ball "images.tar"</i>	tar -cvf images.tar \$(find / -type f -name *.jpg) tar -rvf images.tar \$(find / -type f -name *.jpg) find / -type f -name "*.jpg" -exec tar -cvf images.tar {} \;

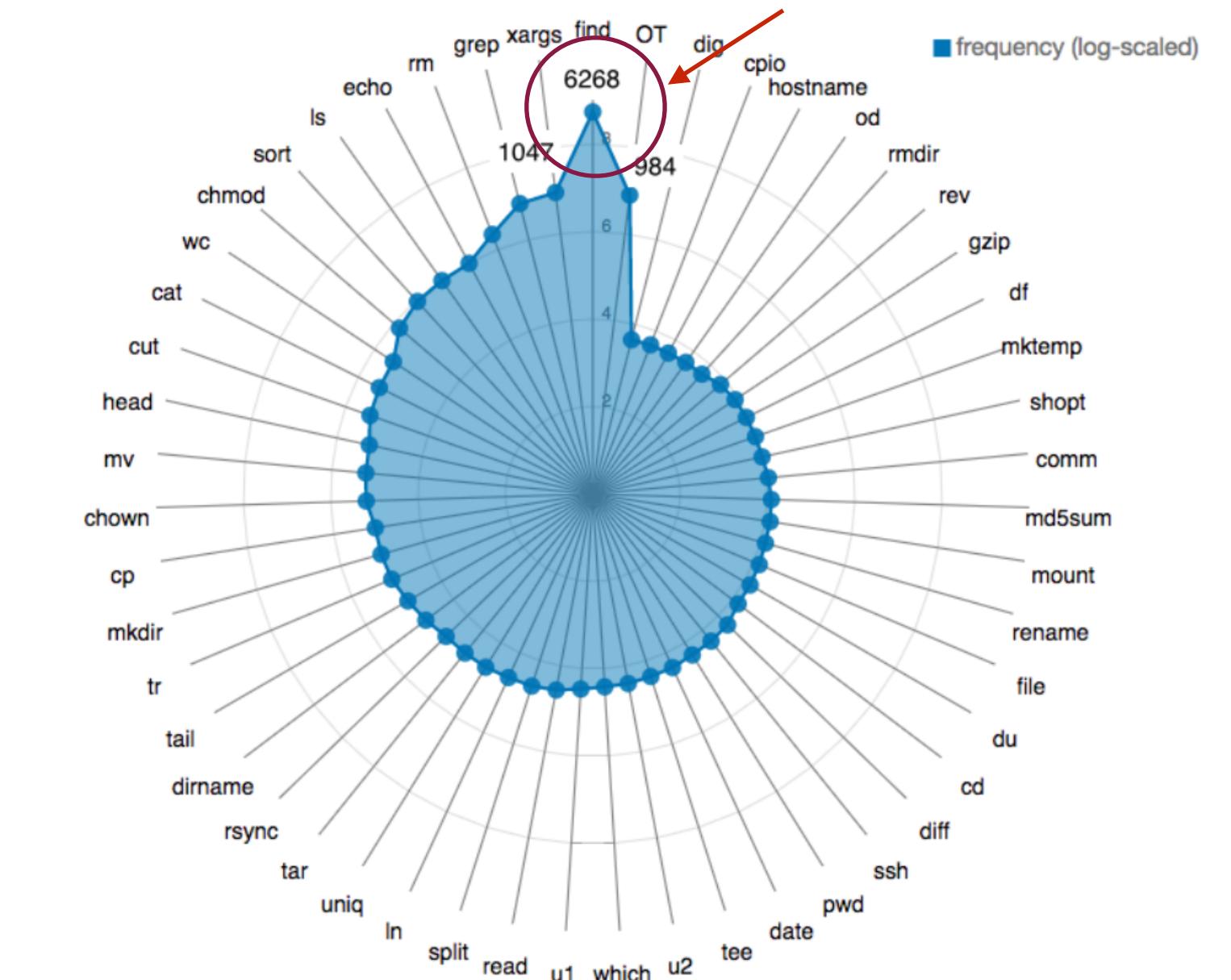


# NL2Bash

By far the largest (~10k pairs) expert-annotated collection of Bash one-liners paired with their natural language descriptions.

Initially started with  
the “find” command

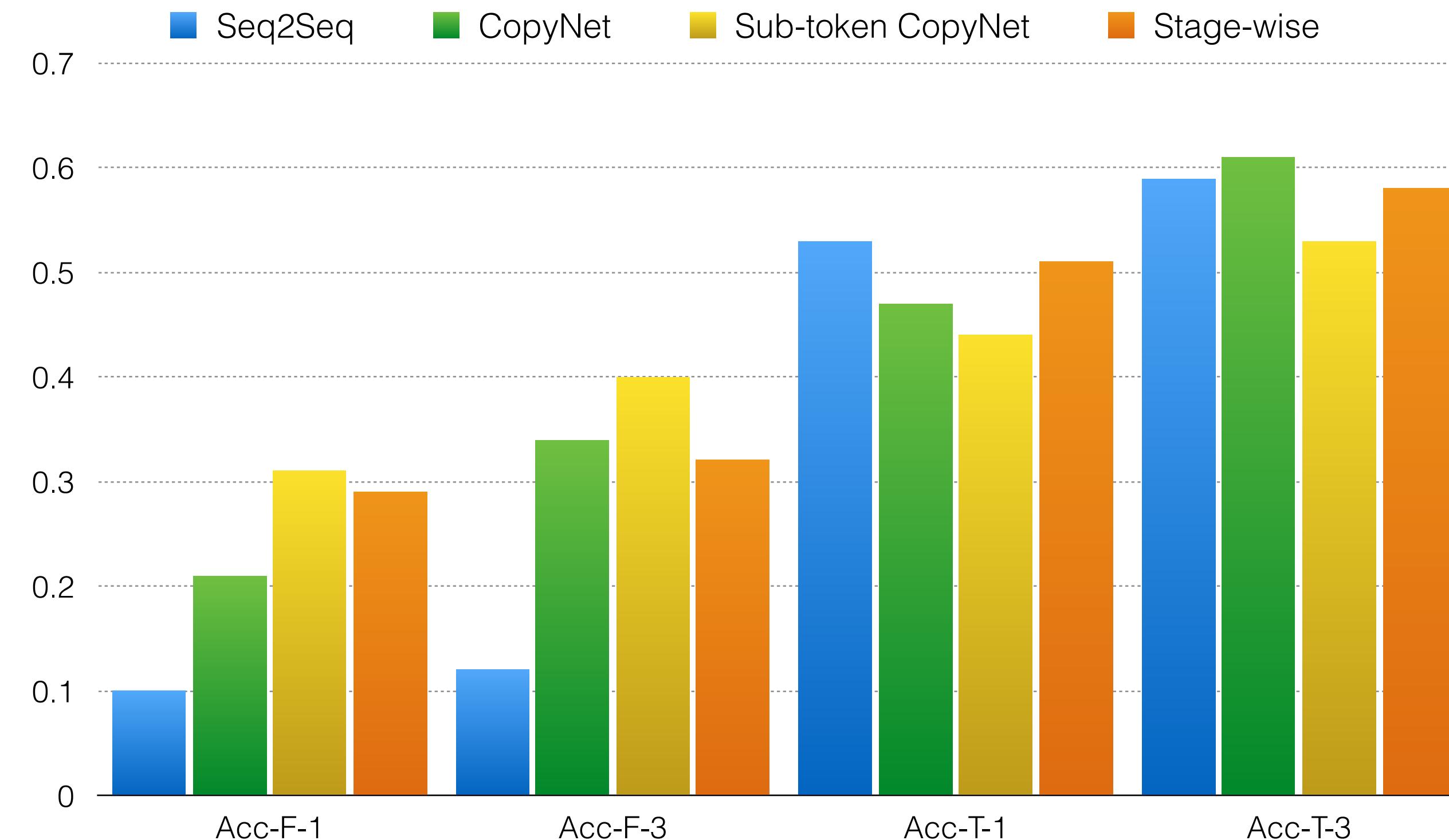
Natural Language	Bash Command(s)
<i>find java files in the current directory tree that contain the pattern ‘TODO’ and print their names</i>	<pre>grep -l "TODO" *.java find . -name "*.java" -exec grep -il "TODO" {} \; find . -name "*.java"   xargs -I {} grep -l "TODO" {}</pre>
<i>display the 5 largest files in the current directory and its sub-directories</i>	<pre>find . -type f   sort -nk 5,5   tail -5 du -a .   sort -rh   head -n5 find . -type f -printf '%s %p\n'   sort -rn   head -n5</pre>
<i>search for all jpg images on the system and archive them to tar ball “images.tar”</i>	<pre>tar -cvf images.tar \$(find / -type f -name *.jpg) tar -rvf images.tar \$(find / -type f -name *.jpg) find / -type f -name "*.jpg" -exec tar -cvf images.tar {} \;</pre>



# Model

CopyNet (Gu et al. 2016)

Evaluation: manual on a subset



# Data Annotation by Experts

## I. Data Annotation by Experts - web page crawling and filtering

Command2NL

Prev Next URL Panel Logout (miccah-castorina)

The screenshot shows the Command2NL interface. At the top, there's a navigation bar with 'Command2NL', 'Prev', 'Next', 'URL Panel', and 'Logout (miccah-castorina)'. Below the navigation is a screenshot of a Stack Overflow search results page for 'Range with leading zero in bash'. The search bar shows 'seq -w 30'. To the right of the screenshot, there's a text annotation: 'Collect all bash commands with the utility seq and the corresponding English description' and 'You may submit other bash commands in the page to earn bonus, but this is not required.' Below the screenshot, the Stack Overflow question details are shown: title 'Range with leading zero in bash', upvotes (13), tags ('bash'), user 'Oleg Razgulyaev', and a note about possible duplicates. On the right side of the Command2NL interface, there's a detailed annotation for the first result: '#1 seq -w 30' with the note 'Print numbers from 1 to 30 with a leading 0'. It also shows a review from 'md jahidul-hamid' with status 'modification-request' and timestamp '2017-07-16T09:24:18.620Z', with a note 'Not all numbers are printed with a leading zero, e.g., 30 is printed as 30.' and three buttons: 'Accept', 'Reject', and 'Change'.

Range with leading zero in bash

How to add leading zero to bash range?  
For example, I need cycle 01,02,03,...,29,30  
How can I implement this using bash?

13 bash share improve this question 3

possible duplicate of [Zero Padding In Bash – Ciro Santilli 刘晓波死 六四事件 法轮功](#) Jul 24 '15 at 16:47

add a comment

asked Nov 14 '12 at 9:48 Oleg Razgulyaev 2,850 ● 2 ● 18 ● 27

#1 seq -w 30  
Print numbers from 1 to 30 with a leading 0

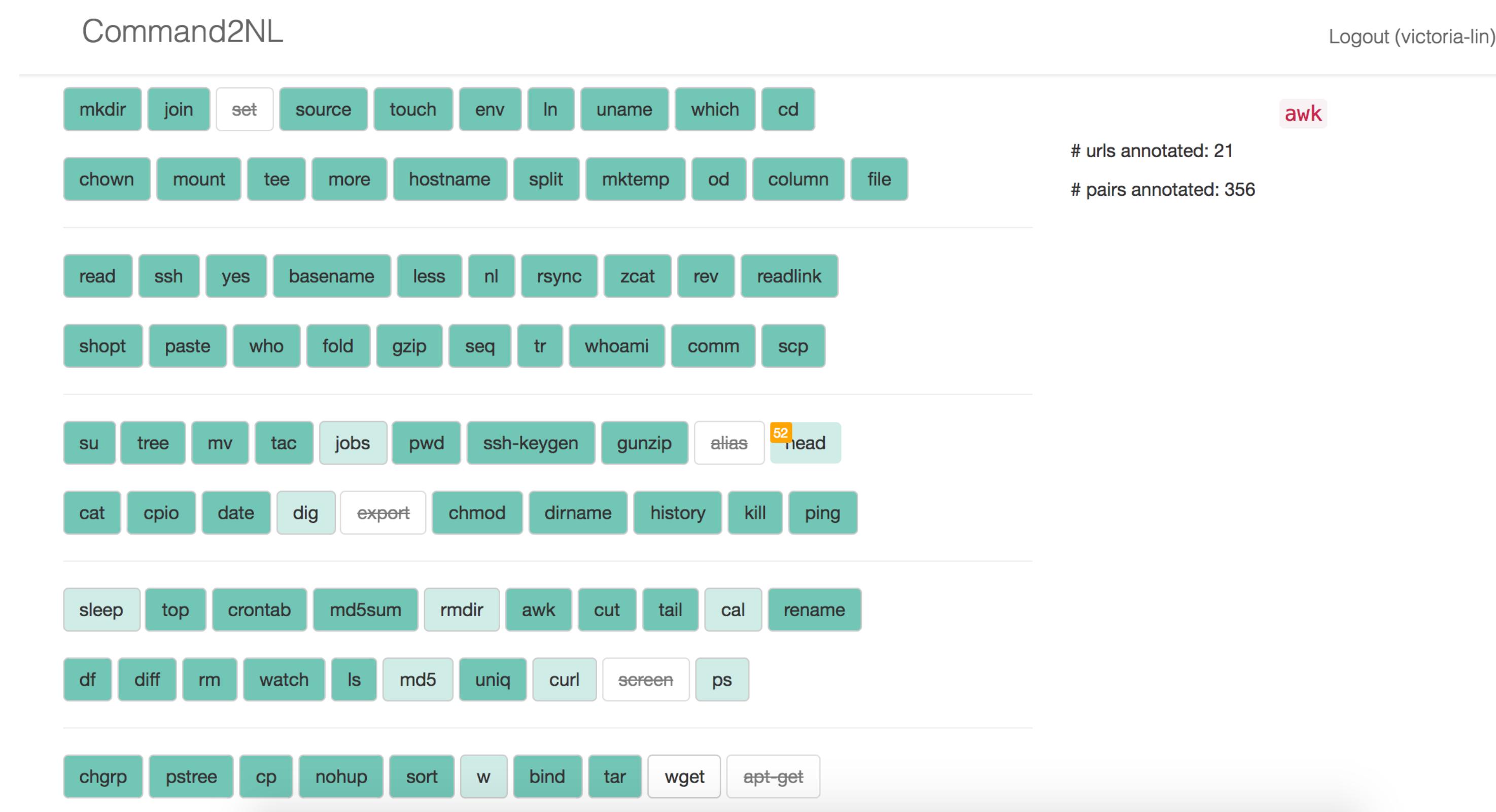
md jahidul-hamid modification-request  
2017-07-16T09:24:18.620Z  
Print numbers from 1 to 30 with equalized 0 padding  
Not all numbers are printed with a leading zero, e.g., 30 is printed as 30.

Accept Reject Change

#2 seq -f "%02g" 30

# Data Annotation by Experts

## II. Data Annotation by Experts - example routing



# Data Annotation by Experts

## III. Data Annotation by Experts - cross validation

Command2NL

Prev Next URL Panel Logout (miccah-castorina)

The screenshot shows a web-based interface for Command2NL. At the top, there's a navigation bar with links for Questions, Developer Jobs, Documentation, Tags, Users, and Log In. Below the navigation, a search bar contains the text "Range with leading zero in bash". To the right of the search bar, there's a note: "Collect all bash commands with the utility seq and the corresponding English description. You may submit other bash commands in the page to earn bonus, but this is not required." A blue dashed oval highlights a specific proposal. The proposal is from user "md jahidul-hamid" (modification-request) dated 2017-07-16T09:24:18.620Z. It suggests changing the command "#1 seq -w 30" to "#2 seq -f "%02g" 30" because "Not all numbers are printed with a leading zero, e.g. 30 is printed as 30." There are three buttons below the proposal: "Accept", "Reject", and "Change". A small arrow points from the text "An expert could propose changes to another expert." towards this proposal area.

Range with leading zero in bash

How to add leading zero to bash range?  
For example, I need cycle 01,02,03,...,29,30  
How can I implement this using bash?

13 bash share improve this question 3

possible duplicate of [Zero Padding In Bash – Ciro Santilli 刘晓波死 六四事件 法轮功](#) Jul 24 '15 at 16:47

asked Nov 14 '12 at 9:48 Oleg Razgulyaev 2,850 ● 2 ● 18 ● 27

#1 seq -w 30  
Print numbers from 1 to 30 with a leading 0

md jahidul-hamid modification-request 2017-07-16T09:24:18.620Z  
Print numbers from 1 to 30 with equalized 0 padding  
Not all numbers are printed with a leading zero, e.g. 30 is printed as 30.

#2 seq -f "%02g" 30

Accept Reject Change

we build reddit-style interaction thread for experts to review the annotations of each other.

An expert could propose changes to another expert. The proposal recipient could accept or reject the change, and the two experts could discuss until they agree with each other.

# Data Annotation by Experts

**However, we observed that the process still have several shortcomings...**

## I. Converting programs to NL descriptions is a very subjective tasks

Even experts tend to omit details that's necessary for computers to make the correct translation, indicating the necessity of modeling ambiguity, omission and pragmatics. (Setlur et al. 2019, 2020)

## II. Annotator Fatigue

Experts stopped producing diverse natural language descriptions after working for a certain amount of time; some came up patterned language to accelerate annotation.

Solution: hiring Amazon Mechanical Turk workers to paraphrase the expert annotation. **Caveat:** Annotating high-quality paraphrases requires expertise.

# Data Annotation by Experts

**However, we observed that the process still have several shortcomings...**

## I. Converting programs to NL descriptions is a very subjective tasks

Even experts tend to omit details that's necessary for computers to make the correct translation, indicating the necessity of modeling ambiguity, omission and pragmatics. (Setlur et al. 2019, 2020)

## II. Annotator Fatigue

Experts stopped producing diverse natural language descriptions after working for a certain amount of time; some came up patterned language to accelerate annotation.

Solution: hiring Amazon Mechanical Turk workers to paraphrase the expert annotation. **Caveat:** Annotating high-quality paraphrases requires expertise.



Focus on collecting high quality evaluation data; modeling ambiguity and pragmatics; creating interaction flow and feedback loop with user. **The training data will never be perfect.**



# Evaluating Code Generation

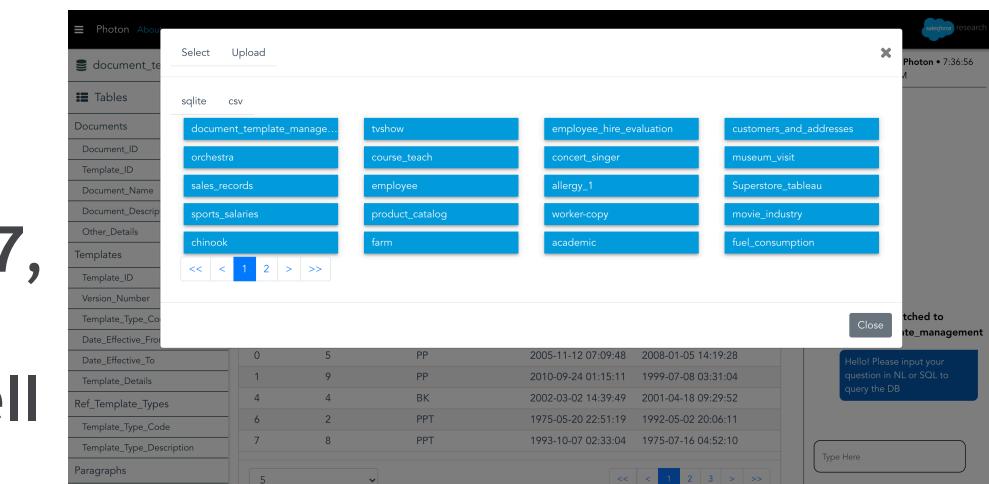
## I. Two pieces of code with different surface forms may have the same semantics.

- Formally verifying program equivalence is expensive.

## II. It's challenging to evaluate Bash code execution results for training and evaluation.

- Safety and security
- Simulate a variety of system environments is non-trivial

 In comparison, evaluation with execution is easy for some NL → code tasks such as NL2SQL (Zhong et al. 2017, Zeng et al. 2020), where the execution environment is well controlled.



# Developing Benchmark Platforms

**It is challenging to carry out such development in academia**

- Difficult to scale
  - Building the systems and platforms costs long student hours
  - Data annotation is expensive
- User study benefits from “real users”



The community could largely benefit from industry open-sourcing such frameworks and open challenges like NLC2CMD.



# Modeling

## Leverage Pre-Trained Language Models

### I. Bash one-liners are great test cases for Seq2Seq (Sutskever et al. 2014) models

- + Copy mechanism for constant values

### II. Pre-trained language models are powerful at handling natural language variations

BERT (Devlin et al. 2018)

GPT-3 (Brown et al. 2020)

BART (Lewis et al. 2020)

T5 (Raffel et al. 2020)

...

 Keep model architecture general in order to generalize to other scripting languages (e.g. Perl, Ruby etc.)



# Modeling

## **III. Pre-training programming language models**

Incorporating External Knowledge through Pre-training for Natural Language to Code Generation. Xu et al. 2020

Unsupervised Translation of Programming Languages. Lachaux et al. 2020

...

## **IV. Incorporating external knowledge such as API documentation, StackOverflow discussion threads, and so on**



# Relieving Data Annotation Demand

## I. Data augmentation

- StaQC:A Systematically Mined Question-Code Dataset from Stack Overflow.Yao et al. 2018
- Data Recombination for Neural Semantic Parsing.Jia & Liang 2016.

## II. Data synthesis

- Schema2QA:Answering Complex Queries on the Structured Web with A Neural Model. Xu et al. 2020
- Genie:A Generator of Natural Language Semantic Parsers for Virtual Assistant Commands. Gampagna et al. 2019

## III. Active Learning & Learning from Interaction

- An Imitation Game for Learning Semantic Parsers from User Interaction.Yao et al. 2020
- Speak to your Parser: Interactive Text-to-SQL with Natural Language Feedback. Elgohary et al. 2020



# Relieving Data Annotation Demand

## I. Data augmentation

- StaQC:A Systematically Mined Question-Code Dataset from Stack Overflow.Yao et al. 2018
- Data Recombination for Neural Semantic Parsing.Jia & Liang 2016.

## II. Data synthesis

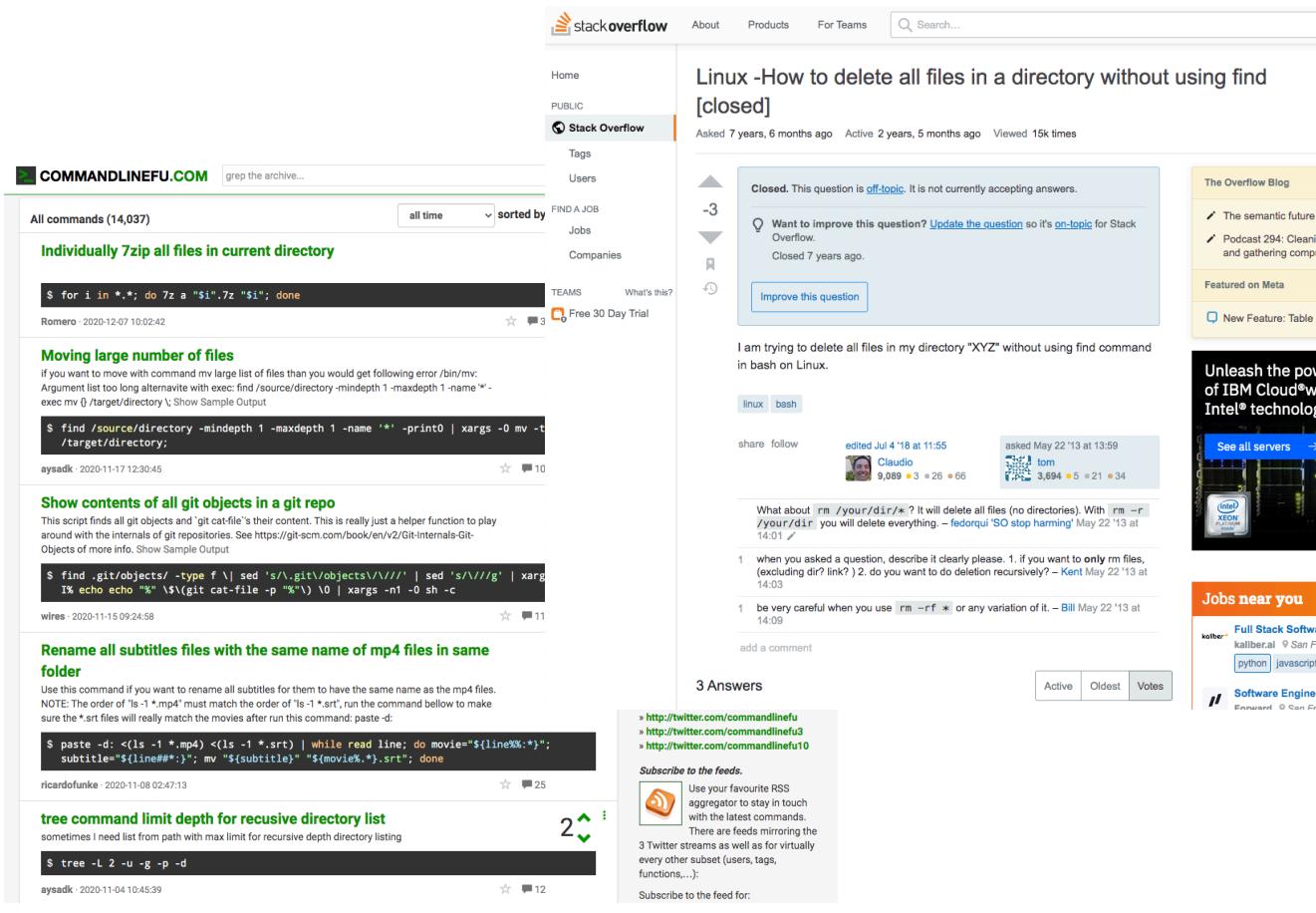
- Schema2SQL: Generating Structured Databases from Natural Language Questions. Xu et al. 2020
- Start from data, but look beyond data
- Genie: A Generator of Natural Language Semantic Parsers for Virtual Assistant Commands. Gampagna et al. 2019

## III. Active Learning & Learning from Interaction

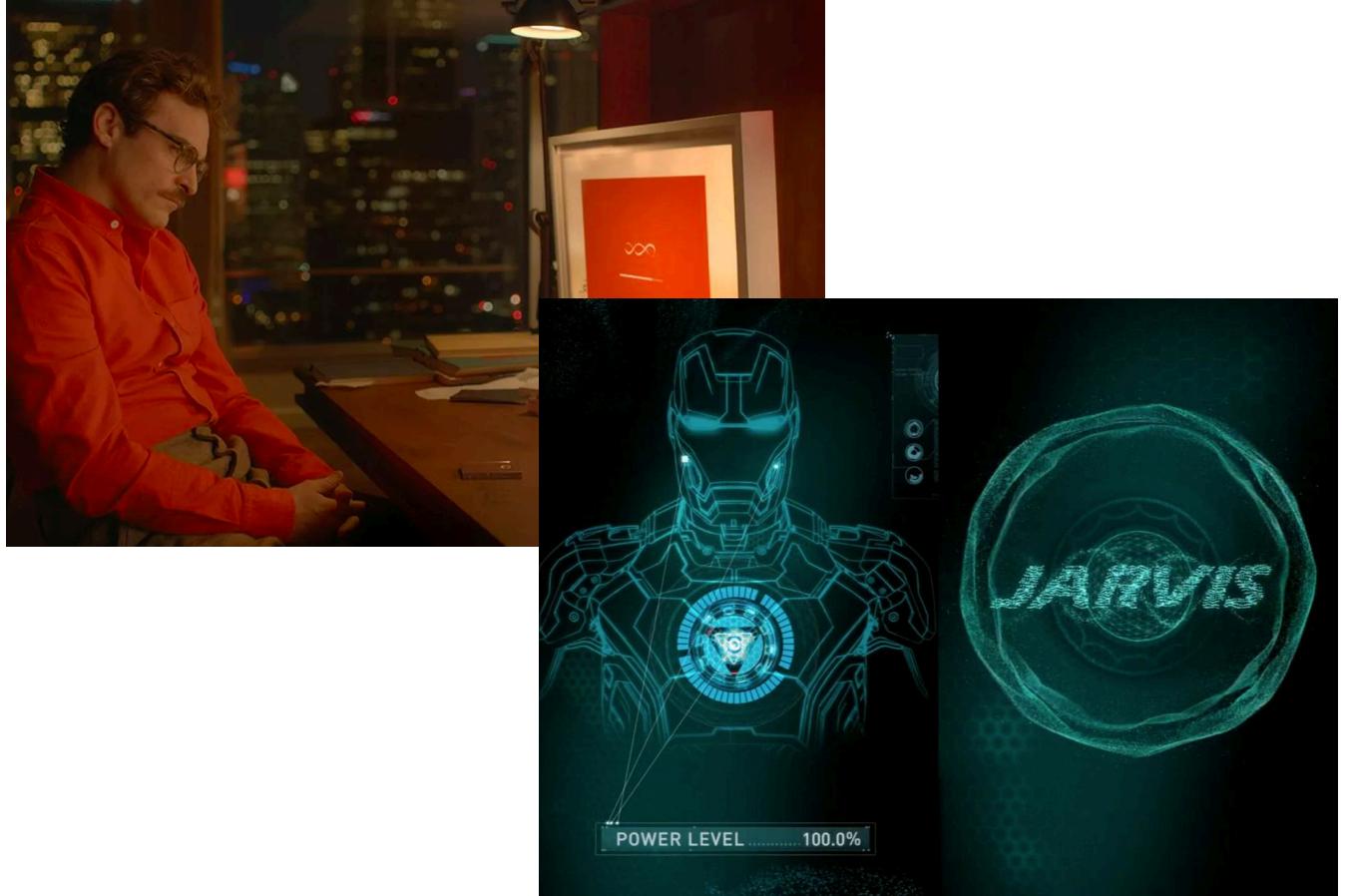
- An Imitation Game for Learning Semantic Parsers from User Interaction.Yao et al. 2020
- Speak to your Parser: Interactive Text-to-SQL with Natural Language Feedback. Elgohary et al. 2020



# The User Control Spectrum



A screenshot of a Stack Overflow question titled "Linux -How to delete all files in a directory without using find [closed]". The question was asked 7 years, 6 months ago and has 14,037 answers. The top answer shows a shell script to 7zip all files in a directory. Below it, there's a section on moving large numbers of files and another on deleting git objects. A sidebar on the right shows a blog post about the semantic future of the web.



**0% machine intelligence**  
**100% user control**

**100% machine intelligence**  
**Very little user control**



# <https://github.com/TellinaTool/tellina>



**Victoria Lin**  
Now at Salesforce



**Chenglong Wang**  
University of Washington



**Deric Pang**  
Now at Google



**Kevin Vu**  
Now at Microsoft



**Luke Zettlemoyer**  
University of Washington



**Mike Ernst**  
University of Washington

1. NL2Bash: A Corpus and Semantic Parser for Natural Language Interface to the Linux Operating System. Lin et al. 2018
2. Program Synthesis from Natural Language Using Recurrent Neural Networks. Lin et al. 2017
3. Natural Language Is A Programming Language: Applying Natural Language Processing to Software Development. Ernst 2017.

**Please reach out to the paper authors if you are interested in contributing to future development!**

