

NL2Bash: A Corpus and Semantic Parser for Natural Language Interface to the Linux Operating System

find system log files
older than a month

```
find / -name "*.log"  
-mtime +30
```

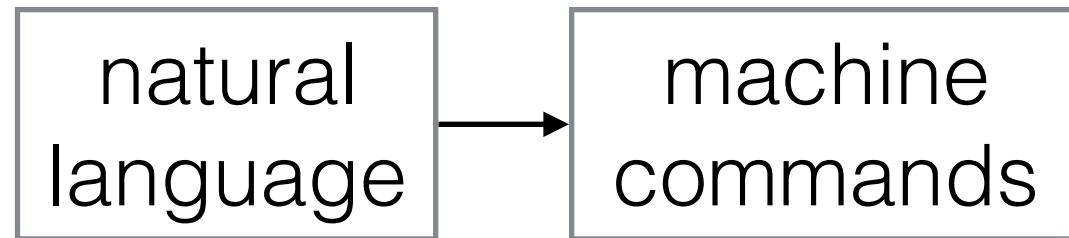
Victoria Lin[☂] Chenglong Wang[☂] Luke Zettlemoyer[☂]

Michael D. Ernst[☂]

{xilin,clwang,lsz,mernst}@cs.washington.edu

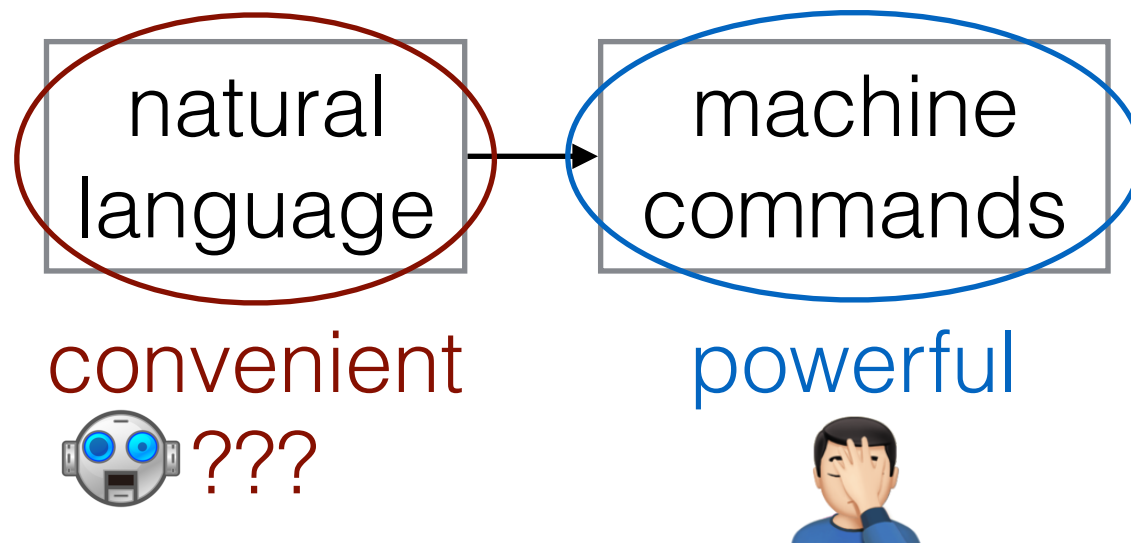
OUTLINE

Problem Definition



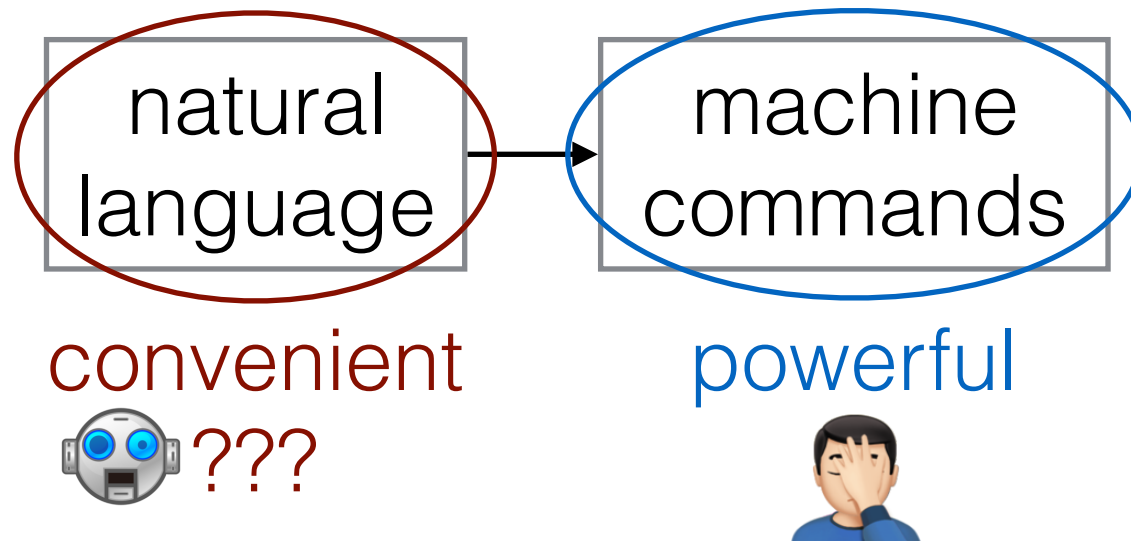
OUTLINE

Problem Definition



OUTLINE

Problem Definition

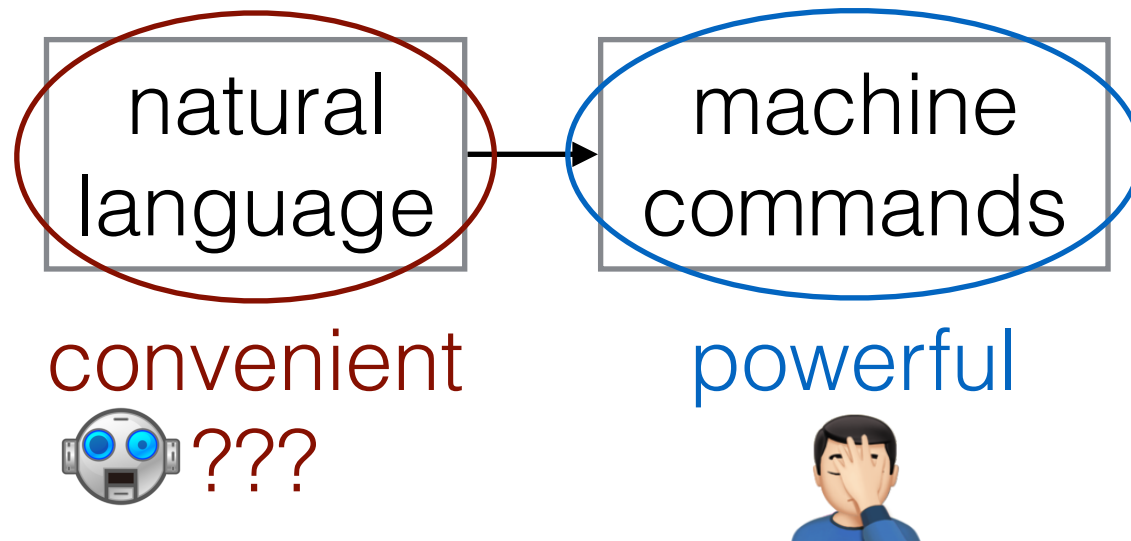


Domain

```
Victorias-MacBook-Pro-2:Projects xilinx$ ls -l
total 0
drwxr-xr-x  22 xilinx staff 748 Jun  8 12:43 helper
drwxr-xr-x   9 xilinx staff 386 Apr 21 2016 opalaye
drwxr-xr-x   7 xilinx staff 236 Jun  9 2016 opalaye_kbp
drwxr-xr-x   5 xilinx staff 170 Dec 17 2015 pigwidgeon
drwxr-xr-x  13 xilinx staff 442 Mar 15 22:47 reflexnet
drwxr-xr-x  12 xilinx staff 486 Jun  6 14:14 resume
drwxr-xr-x  26 xilinx staff 884 Dec 18 2015 skim
drwxr-xr-x  22 xilinx staff 748 Jun  9 2016 toploc2
drwxr-xr-x  31 xilinx staff 1054 Feb 21 14:13 task_platform
drwxr-xr-x  16 xilinx staff 544 Mar 29 18:18 tellino
drwxr-xr-x  52 xilinx staff 1768 May 10 20:44 tellino_fsm
drwxr-xr-x  11 xilinx staff 374 Mar 31 21:02 todopole3.github.io
drwxr-xr-x  62 xilinx staff 2188 Oct  9 2015 tutor
dmesg-----@ 16 xilinx staff 544 Feb 12 12:13 sysfs-ops
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py"
./reflexnet/net/beam_search.py
./tellino/tellino_learning_module/encoder_decoder/beam_search.py
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
./reflexnet/net/beam_search.py
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
```

OUTLINE

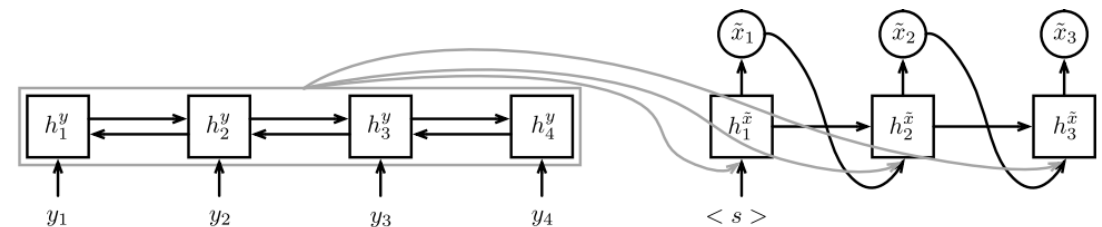
Problem Definition



Domain

```
Victorias-MacBook-Pro-2:Projects xilinx$ ls -l
total 0
drwxr-xr-x  22 xilinx staff 748 Jun 8 12:43 helper
drwxr-xr-x   9 xilinx staff 386 Apr 21 2016 opaloye
drwxr-xr-x   7 xilinx staff 236 Jun 9 2016 opaloye_kbp
drwxr-xr-x   5 xilinx staff 170 Dec 17 2015 pigwidgeon
drwxr-xr-x  13 xilinx staff 442 Mar 15 22:47 reflexnet
drwxr-xr-x  12 xilinx staff 486 Jun 6 14:14 resume
drwxr-xr-x  26 xilinx staff 884 Dec 18 2015 skim
drwxr-xr-x  22 xilinx staff 748 Jun 9 2016 toploc
drwxr-xr-x  31 xilinx staff 1854 Feb 21 14:13 task_platform
drwxr-xr-x  16 xilinx staff 544 Mar 29 18:18 tellino
drwxr-xr-x  52 xilinx staff 1768 May 10 20:44 tellino_fsm
drwxr-xr-x  11 xilinx staff 374 Mar 31 21:02 todopolis.github.io
drwxr-xr-x  62 xilinx staff 2186 Oct 9 2015 tutor
drwxr-xr-x  16 xilinx staff 544 Feb 12 12:13 xilinx-ops
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py"
./reflexnet/net/beam_search.py
./tellino/tellino_learning_module/encoder_decoder/beam_search.py
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
./reflexnet/net/beam_search.py
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
```

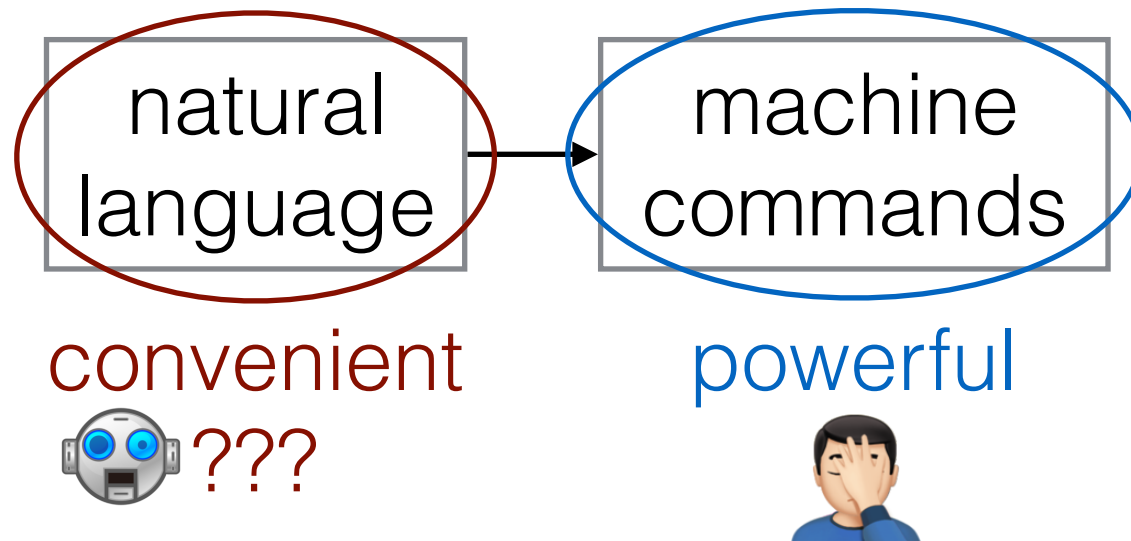
Data-Driven Approaches



Adaptions from state-of-the-art neural machine translation models

OUTLINE

Problem Definition



Corpus Construction



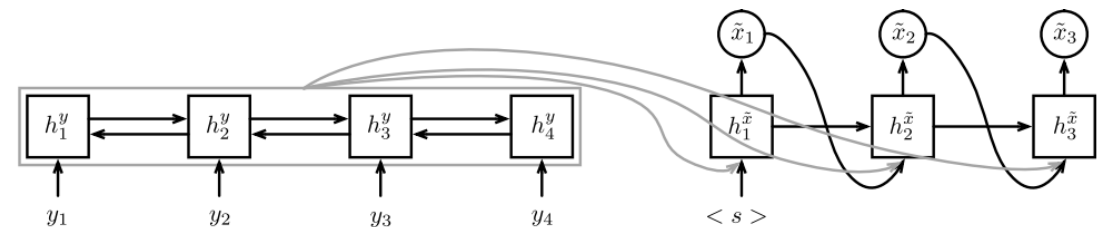
Domain

```

Victoria's-MacBook-Pro-2:Projects killing %1
total 0
dwxor-xr-x  22 xlin staff  748 Jan  8 2016 hopsy
dwxor-xr-x  9 xlin staff  306 Apr 21 2016 opolaye
dwxor-xr-x  7 xlin staff  238 Jun  9 2016 opolaye_kbp
dwxor-xr-x  5 xlin staff  170 Dec 17 2015 pigtdgeon
dwxor-xr-x 13 xlin staff  442 Mar 15 22:47 refnetct
dwxor-xr-x 12 xlin staff  408 Jun  6 14:14 resume
dwxor-xr-x 26 xlin staff  886 Dec 18 2015 skm
dwxor-xr-x 22 xlin staff  748 Jun  9 2016 topoc
dwxor-xr-x 31 xlin staff 1054 Feb 21 14:13 task_platform
dwxor-xr-x 16 xlin staff 544 Mar 29 18:18 tellina
dwxor-xr-x 32 xlin staff 1703 Nov 18 20:45 teller_fie
dwxor-xr-x 11 xlin staff 374 Mar 31 21:82 topodocs.github.io
dwxor-xr-x 62 xlin staff 2188 Oct  9 2015 tour
dwxor-xr-x 16 xlin staff 544 Sep 11 2015 type-soc
Victoria's-MacBook-Pro-2:Projects killing find -name "beam_search.py"
./helper/encoder_decoder/beam_search.py
./reflexion/net/beam_search.py
./reflexion/learning_module/encoder_decoder/beam_search.py
Victoria's-MacBook-Pro-2:Projects killing find -name "beam_search.py" -mtime -1
./helper/encoder_decoder/beam_search.py
Victoria's-MacBook-Pro-2:Projects killing find all files named "beam_search.py"
that was modified today

```

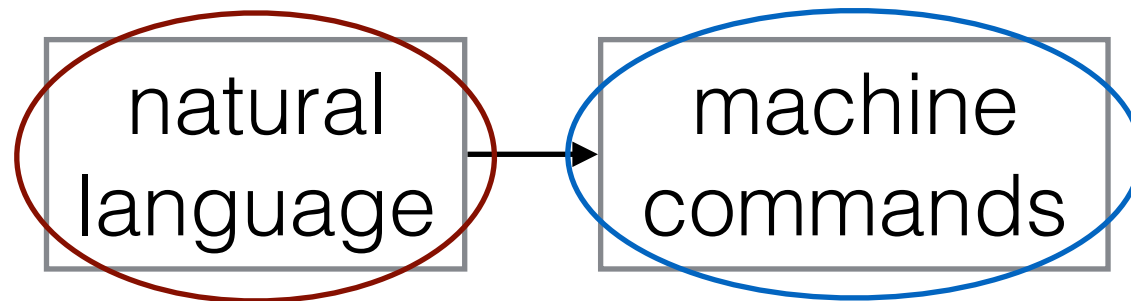
Data-Driven Approaches



Adaptions from state-of-the-art neural machine translation models

OUTLINE

Problem Definition



convenient



powerful



Corpus Construction

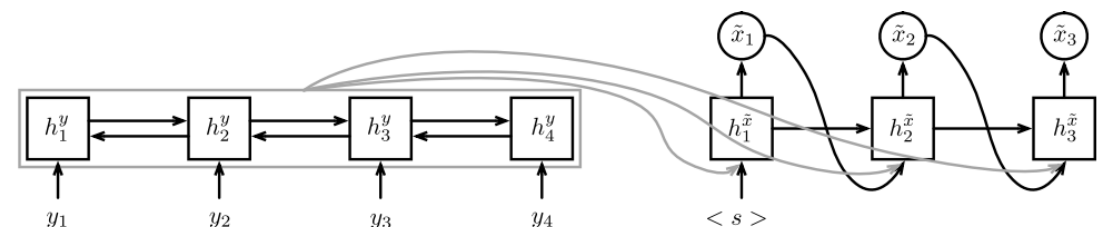


Domain

```
Victoria-MacBook-Pro-2:Projects xilin$ ls -l
total 0
drwxr-xr-x  22 xilin staff 748 Jun  8 12:43 helper
drwxr-xr-x   9 xilin staff 386 Apr 21 2016 opaloye
drwxr-xr-x   7 xilin staff 236 Jun  9 2016 opaloye_kbp
drwxr-xr-x   5 xilin staff 170 Dec 17 2015 pigwidgeon
drwxr-xr-x  13 xilin staff 442 Mar 15 22:47 reflexnet
drwxr-xr-x  12 xilin staff 486 Jun  6 14:14 resume
drwxr-xr-x  26 xilin staff 884 Dec 18 2015 skim
drwxr-xr-x  22 xilin staff 748 Jun  9 2016 toploc
drwxr-xr-x  31 xilin staff 1854 Feb 21 14:13 task_platform
drwxr-xr-x  16 xilin staff 544 Mar 29 18:18 tellino
drwxr-xr-x  52 xilin staff 1768 May 10 20:44 tellino_fsm
drwxr-xr-x  11 xilin staff 374 Mar 31 21:08 todopolis.github.io
drwxr-xr-x  62 xilin staff 2186 Oct  9 2015 tutor
drwxr-xr-x  16 xilin staff 544 Feb 12 12:12 xps-asp

Victoria-MacBook-Pro-2:Projects xilin$ find -name "beam_search.py"
./reflexnet/net/beam_search.py
./tellino/tellino_learning_module/encoder_decoder/beam_search.py
Victoria-MacBook-Pro-2:Projects xilin$ find -name "beam_search.py" -mtime -1
./reflexnet/net/beam_search.py
Victoria-MacBook-Pro-2:Projects xilin$ find -name "beam_search.py" -mtime -1
Victoria-MacBook-Pro-2:Projects xilin$
```

Data-Driven Approaches



Adaptions from state-of-the-art neural machine translation models

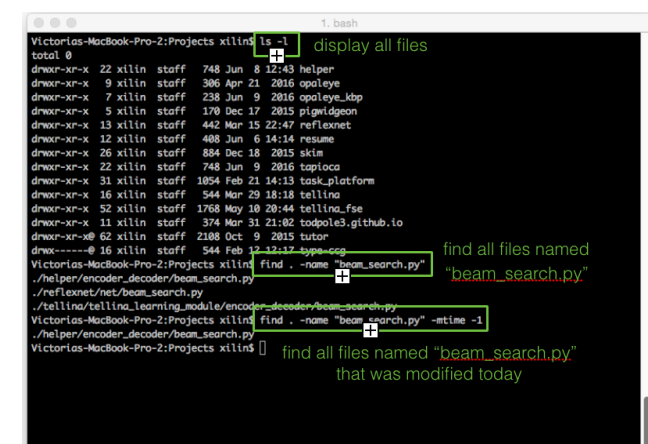
System Performance
Qualitative Analysis
Live Demo

PROBLEM DEFINITION

- Natural Language → Command Translation
 - Generating **one-liners**
 - In most command languages complex semantics can be represented in short syntactic forms
 - Other work: code block generation (Polosukhin and Skidanov '18)
 - **Single-turn interaction** between the user & the system (building block for multi-turn system)
 - Other work: conversational natural language programming assistant (Pandita et. al. '18)
 - Semantic parsing can be a building block conversational programming assistant

DOMAIN - BASH

- Potentially Wide User Base
 - Most Linux users know bash, but not mastering it
- Command Interface Language
- Generalizable to other command languages



```
Victorias-MacBook-Pro-2:Projects xilinx$ ls -l
total 0
drwxr-xr-x  22 xilin  staff   748 Jun  8 12:43 helper
drwxr-xr-x   9 xilin  staff   306 Apr 21 2016 opaleye
drwxr-xr-x   7 xilin  staff   238 Jun  9 2016 opaleye_kbp
drwxr-xr-x   5 xilin  staff   170 Dec 17 2015 pigwidgeon
drwxr-xr-x  13 xilin  staff   442 Mar 15 22:47 reflexnet
drwxr-xr-x  12 xilin  staff   408 Jun  6 14:14 resume
drwxr-xr-x  26 xilin  staff   884 Dec 18 2015 skim
drwxr-xr-x  22 xilin  staff   748 Jun  9 2016 tapioca
drwxr-xr-x  31 xilin  staff  1054 Feb 21 14:13 task_platform
drwxr-xr-x  16 xilin  staff   544 Mar 29 18:18 tellina
drwxr-xr-x  52 xilin  staff  1768 May 10 20:44 tellina_fsm
drwxr-xr-x  11 xilin  staff   374 Mar 31 21:02 todpole3.github.io
drwxr-xr-x@ 62 xilin  staff  2188 Oct  9 2015 tutor
drwxr-xr-x@ 16 xilin  staff   544 Feb 12 12:17 type-csg
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py"
./reflexnet/net/beam_search.py
./tellina/tellina_learning_module/encoder_decoder/beam_search.py
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
./helper/encoder_decoder/beam_search.py
Victorias-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
find all files named "beam_search.py" that was modified today
```

BASH EXAMPLE

- find all '*.c' files under \$HOME directory which contain the string "Salesforce"

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"Salesforce" {}
```

BASH EXAMPLE

- find all '*.c' files under \$HOME directory which contain the string "Salesforce"

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"Salesforce" {}
```

Head command

BASH EXAMPLE

- find all '*.c' files under \$HOME directory which contain the string "Salesforce"

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"Salesforce" {}
```

Flag

BASH EXAMPLE

- find all '*.c' files under \$HOME directory which contain the string "Salesforce"

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"Salesforce" {}
```

Argument

BASH EXAMPLE

- find all '*.c' files under \$HOME directory which contain the string "Salesforce"

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"Salesforce" {}
```

Compound Commands

RELATED WORK

- Neural Networks: Natural Language → Formal Languages
 - ✓ NL → Syntactic parse trees (Vinyals et. al. '14)
 - ✓ NL → Regular expression (Locascio et. al. '16)
 - ✓ NL → Logical forms (Li & Lapata '16)
 - ✓ NL → Python (Wang et. al. '16)
 - ✓ NL → Python (Yin & Neubig '17, Rabinovich et. al. '17)

Rule-Based
Systems

Statistical Models over
Discrete Structures

RELATED WORK

- Neural Networks: Natural Language → Formal Languages
 - ✓ NL → Syntactic parse trees (Vinyals et. al. '14)
 - ✓ NL → Regular expression (Locascio et. al. '16)
 - ✓ NL → Logical forms (Li & Lapata '16)
 - ✓ NL → Python (Wang et. al. '16)
 - ✓ NL → Python (Yin & Neubig '17, Rabinovich et. al. '17)

Adapted from NMT methods for natural language translation

RELATED WORK

- Neural Networks: Natural Language → Formal Languages

✓ NL → Syntactic parse trees (Vinyals et. al. '14)

✓ NL → Regular expression (Locascio et. al. '16)

✓ NL → Logical forms (Li & Lapata '16)

✓ NL → Python (Wang et. al. '16)

✓ NL → Python (Yin & Neubig '17, Rabinovich et. al. '17)

Seq2Seq

Seq2Tree

Expressive —> Simplest Data Representation

RELATED WORK

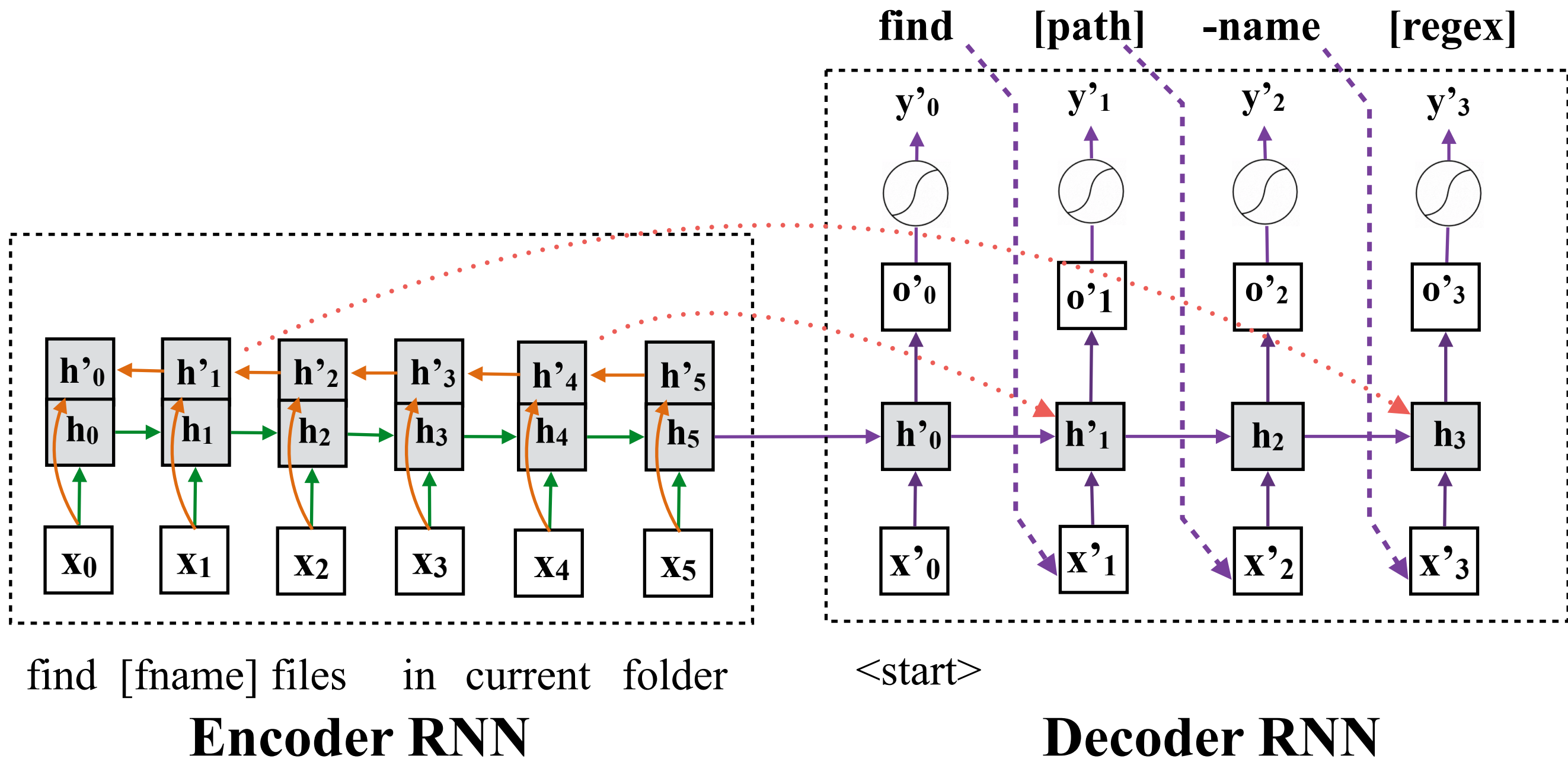
- Neural Networks: Natural Language → Formal Languages

- ✓ NL → Syntactic parse trees (Vinyals et. al. '14)
- ✓ NL → Regular expression (Locascio et. al. '16)
- ✓ NL → Logical forms (Li & Lapata '16)
- ✓ NL → Python (Wang et. al. '16)

Seq2Seq

Target Domain: Shallow Syntax Structure (No Formal Grammar)

SEQUENCE-TO-SEQUENCE NEURAL NETWORK



SEQ2SEQ + COPYING

- find all **'*.c'** files under **\$HOME** directory whose content has the string **"salesforce"**

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"salesforce" {}
```

✗ Large number of out-of-vocabulary words (arguments)

SEQ2SEQ + COPYING

- find all **'*.c'** files under **\$HOME** directory whose content has the string **"salesforce"**

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"salesforce" {}
```

× Large number of out-of-vocabulary words

SEQ2SEQ + COPYING

- find all **'*.c'** files under **\$HOME** directory whose content has the string **"salesforce"**

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"salesforce" {}
```

- ✗ Many command arguments are source tokens transformed through atomic string edits

SEQ2SEQ + COPYING

- find all **'*.c'** files under **\$HOME** directory whose content has the string **"salesforce"**

```
find "$HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"salesforce" {}
```

- ✗ Many command arguments are source tokens transformed through atomic string edits

Character models? Very long sequences...

SUB-TOKEN COPYING

- find all '***.c**' files under **\$ HOME** directory whose content has the string "**salesforce**"

```
find "$ HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"salesforce" {}
```

Split the constant tokens in both the source and target sequences into a sequence of sub-tokens consists of the following:

1. Consecutive sub-sequences of alphabetical letters
2. Consecutive sub-sequences of digits
3. All other special tokens

Run CopyNet on the sub-tokens

SUB-TOKEN COPYING

- find all '***.c**' files under **\$ HOME** directory whose content has the string "**salesforce**"

```
find "$ HOME" -name "*.c" -print0 | xargs -0 -I {} grep  
"salesforce" {}
```

Enables learning of

1. Substring addition
2. Substring deletion
3. Substring replacement
4. Semantics of the special characters such as "\$", quotation marks, "*", etc.

DATA COLLECTION

- Bash programmers hired **upwork**TM
- Collect bash commands and their natural language descriptions from the web



✓ web interface to control the collection process

BASH COMMAND FILTERING

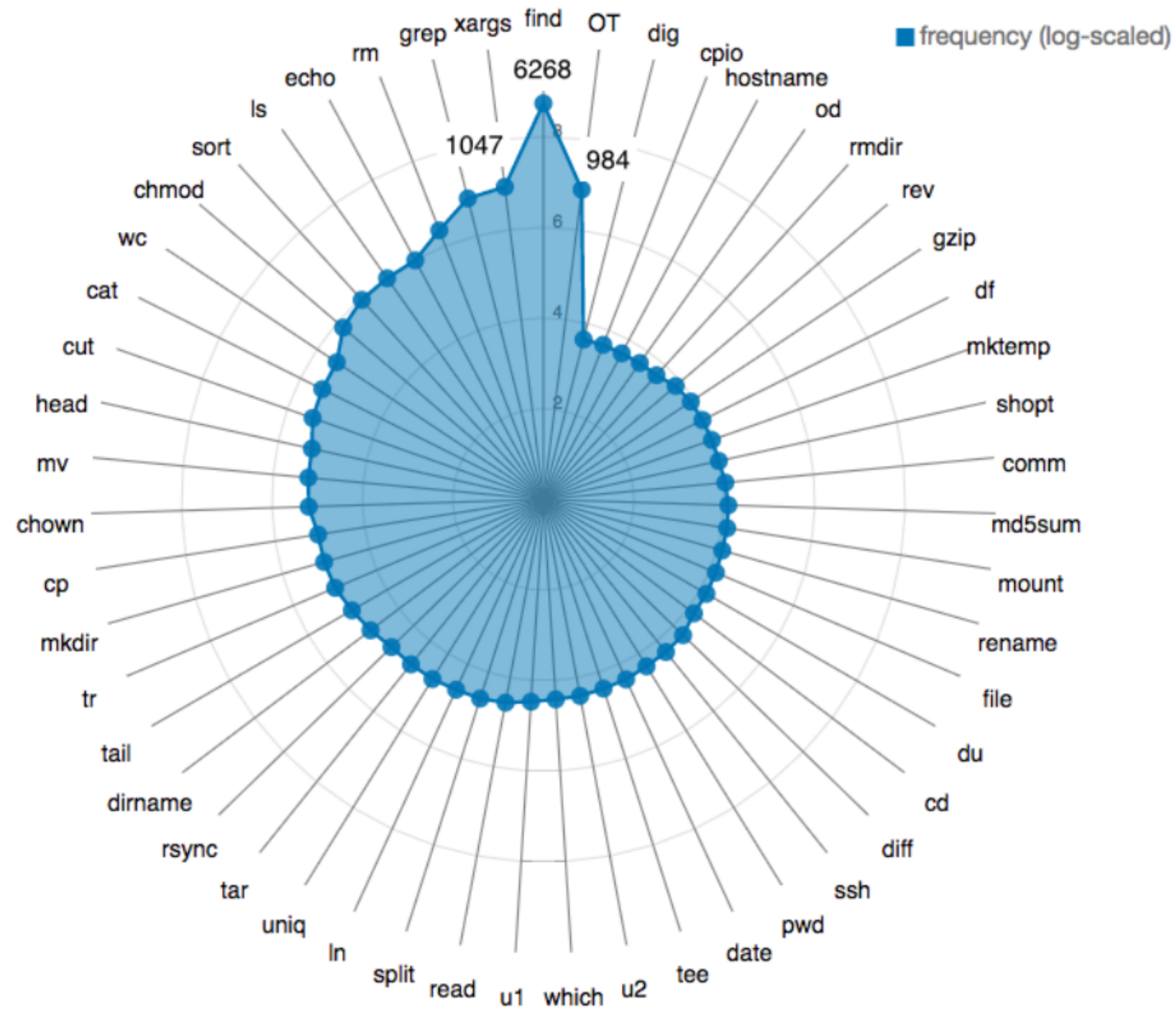
- Bash Command

In-scope	Single command	
	Logical connectives	&&, , ()
		pipeline
	Nested command	command substitution \$() process substitution <()
Out-of-scope	I/O redirection	<, <<
	Variable assignment	=
	Parameters	e.g. \$1, \$HOME
	Multi-statement	if, for, while, until, etc.
	Regex structure	e.g. x*y*
	Non-bash programs	triggered by awk, java, etc.

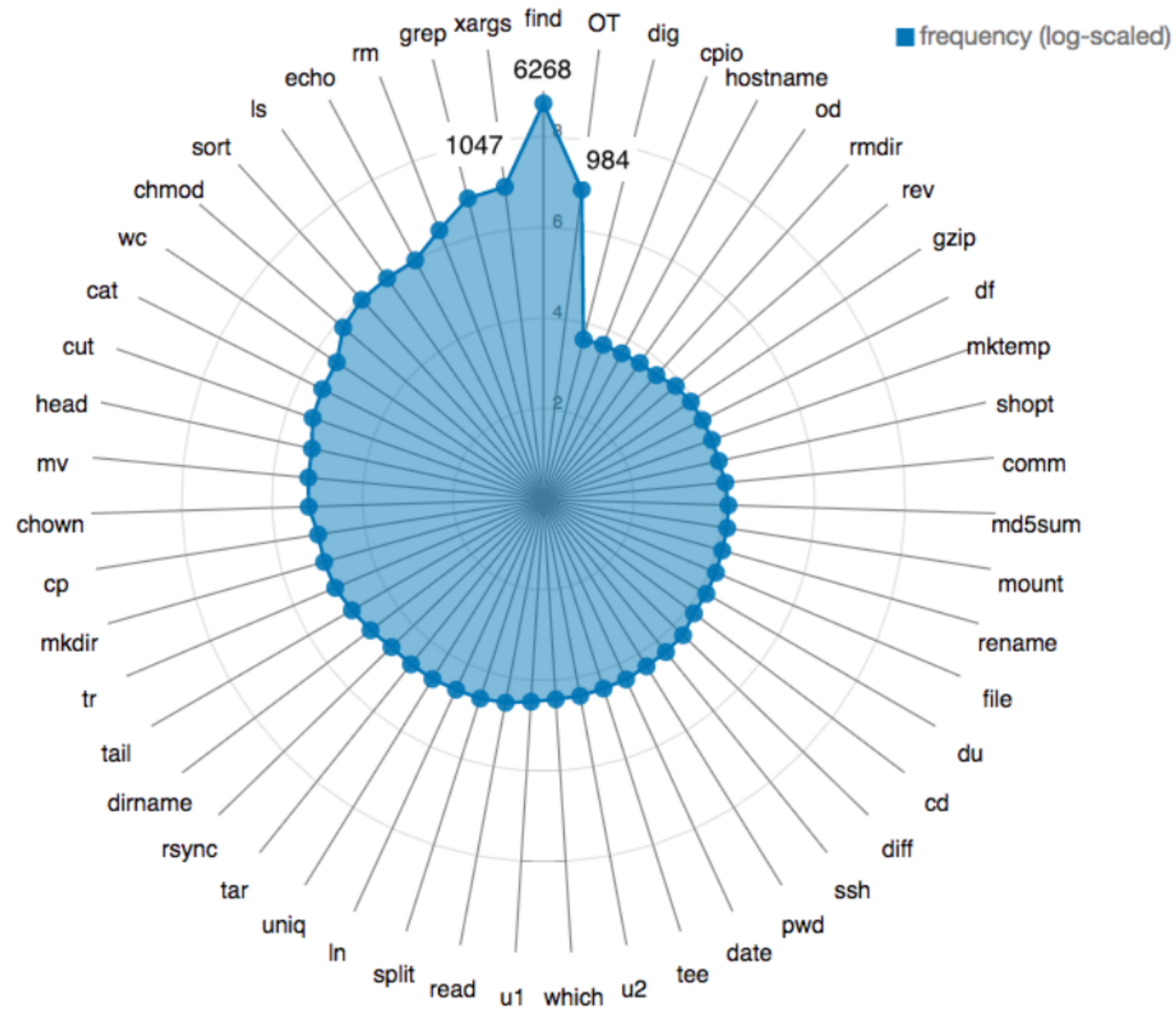
DATA STATISTICS

- 12,609 pairs —> 9,301 pairs after filtering
- 8,090 train, 609 dev, 606 test
- 100+ unique bash commands, 537 unique flags

TOP-50 COMMAND HISTOGRAM



TOP-50 COMMAND HISTOGRAM



The rest combined: 984

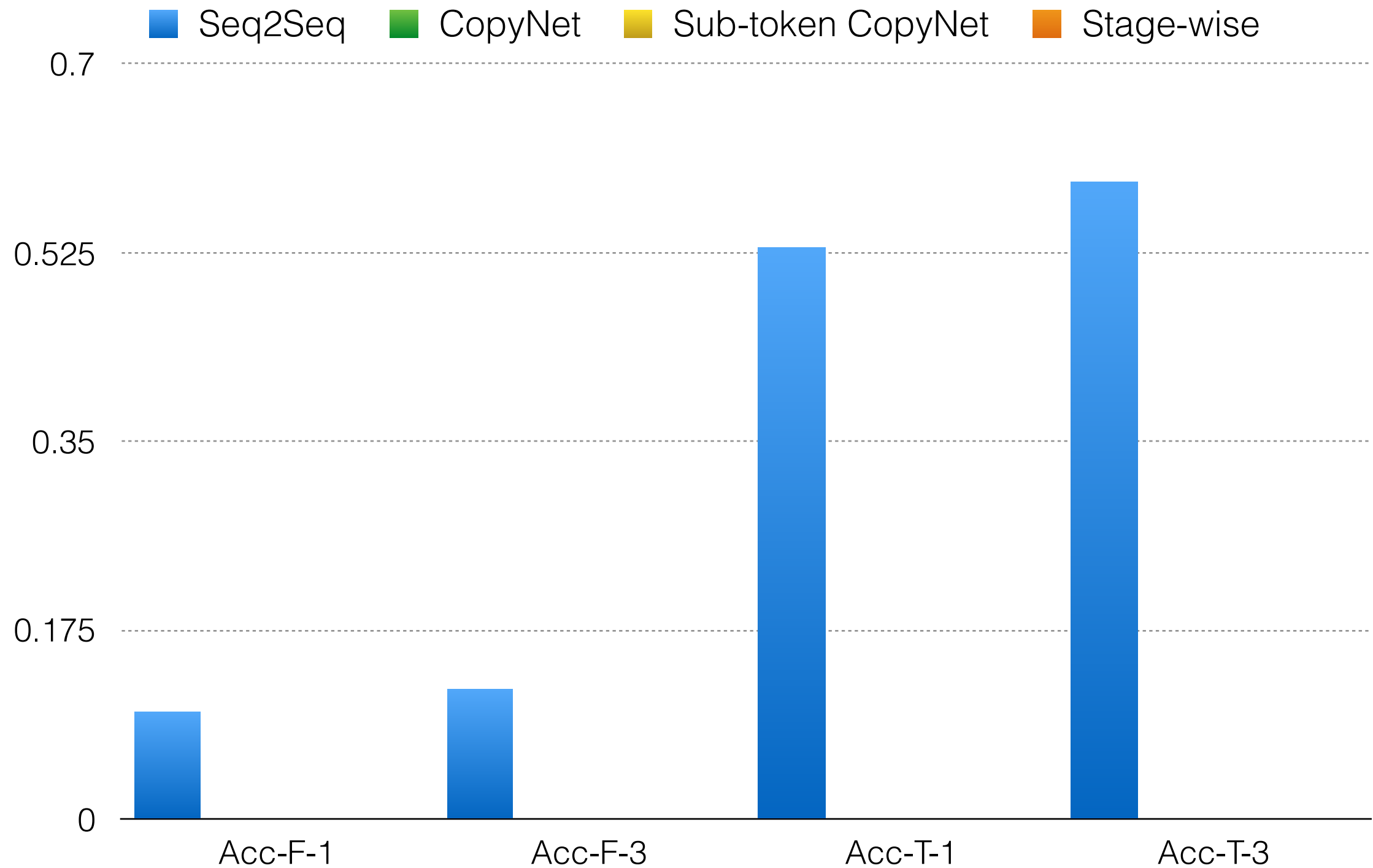
EVALUATION METHODOLOGY

- Manual Evaluation (Multiple Correct Solutions)
 - 3 bash programmers (hired via **upwork**[™]) judged the top-3 suggestions of each test example
 - Full command correctness
 - Command template correctness
- find [path] -name [regex] -print0 | xargs -0 -I {} grep [regex] {}**
- Final judgement: majority vote
 - Inter-annotator agreement: 0.89, 0.83, 0.80

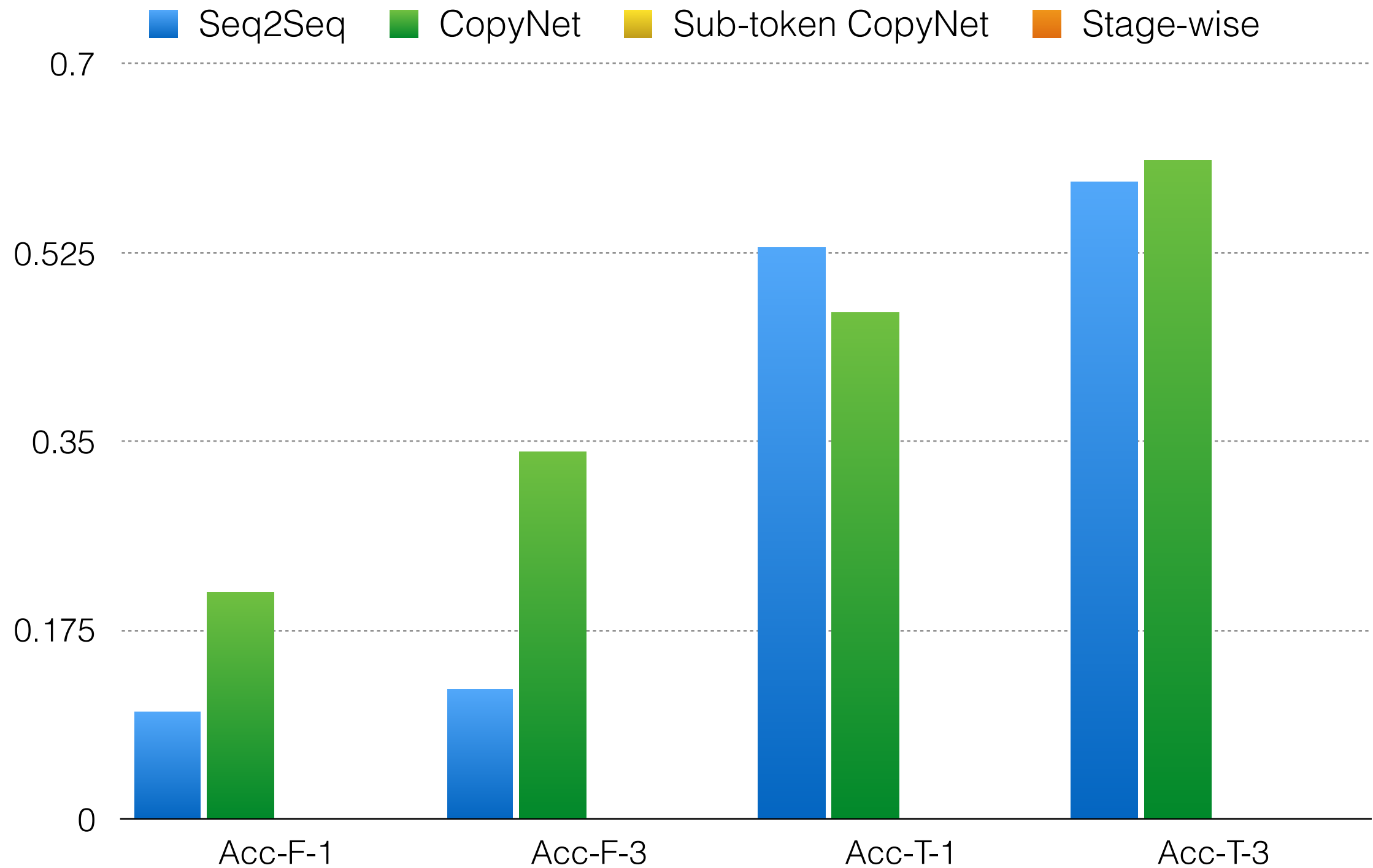
BASELINES

- Vanilla Seq2Seq (Sutskever et. al. '14)
- CopyNet (Gu et. al. '17)
- Three-stage translation model (Lin et. al. '17)
 1. Convert both NL and bash command to templates
 2. Apply Seq2Seq translation on the templates
 3. Fill arguments using heuristics

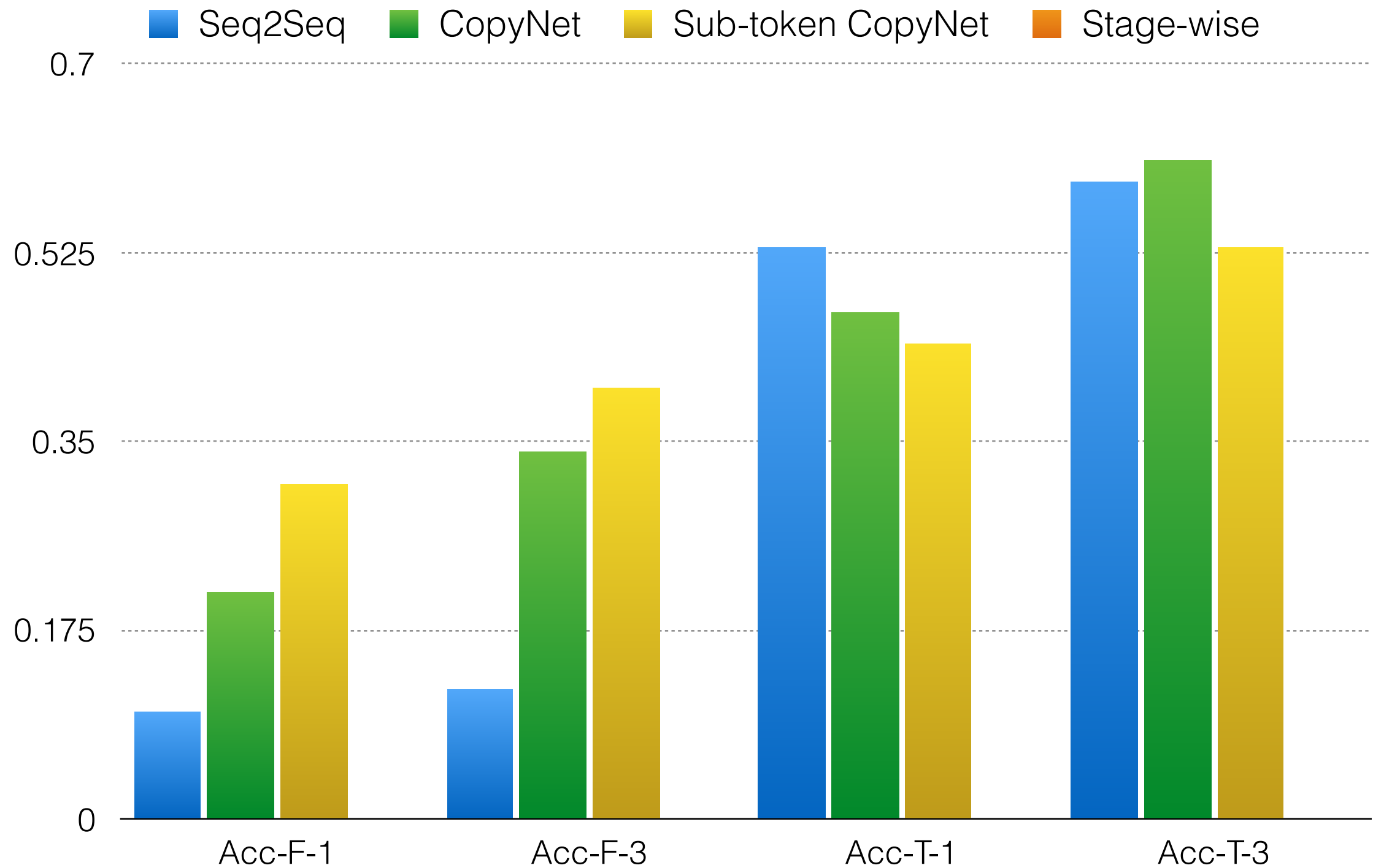
SYSTEM PERFORMANCE (Dev Set)



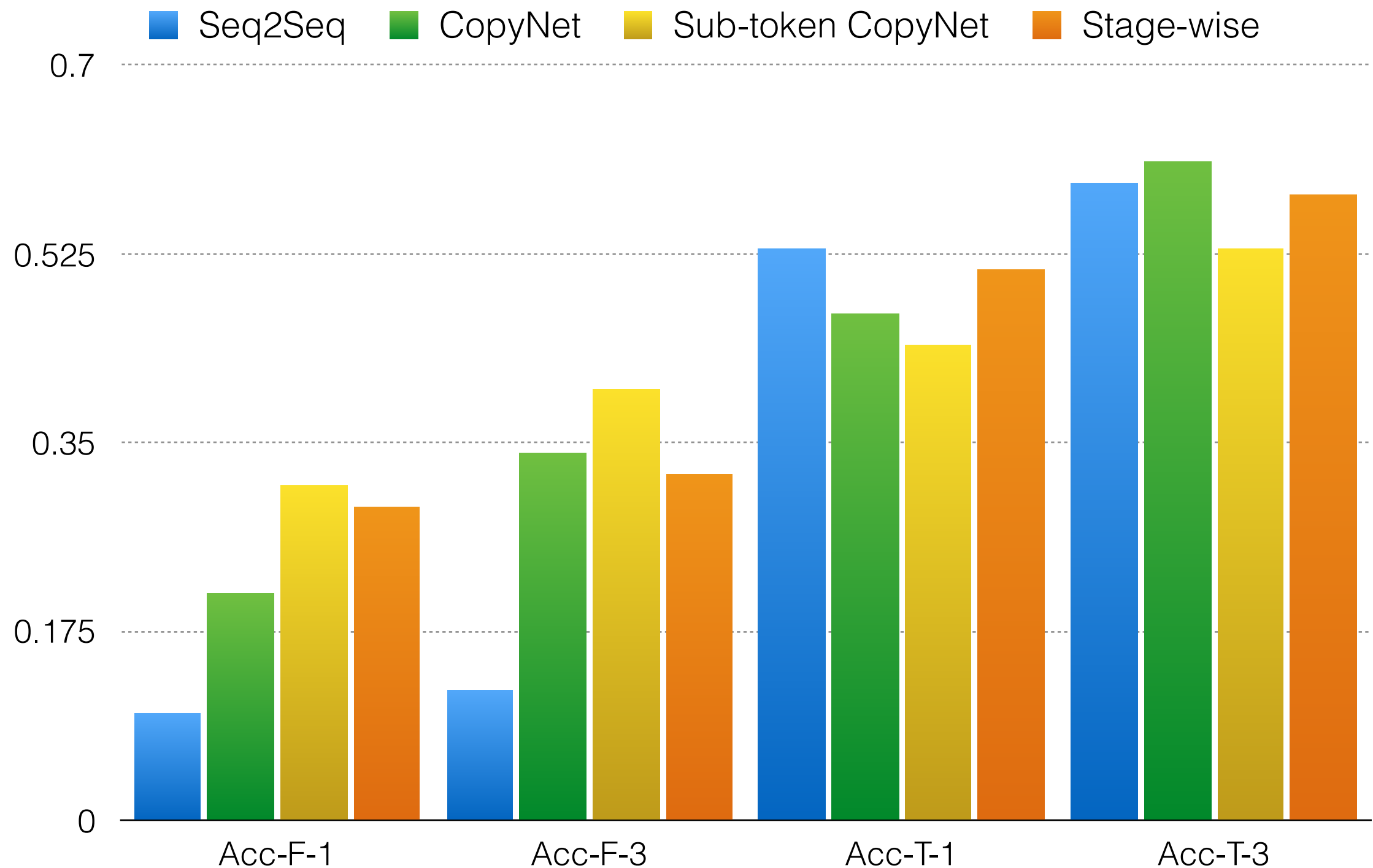
SYSTEM PERFORMANCE (Dev Set)



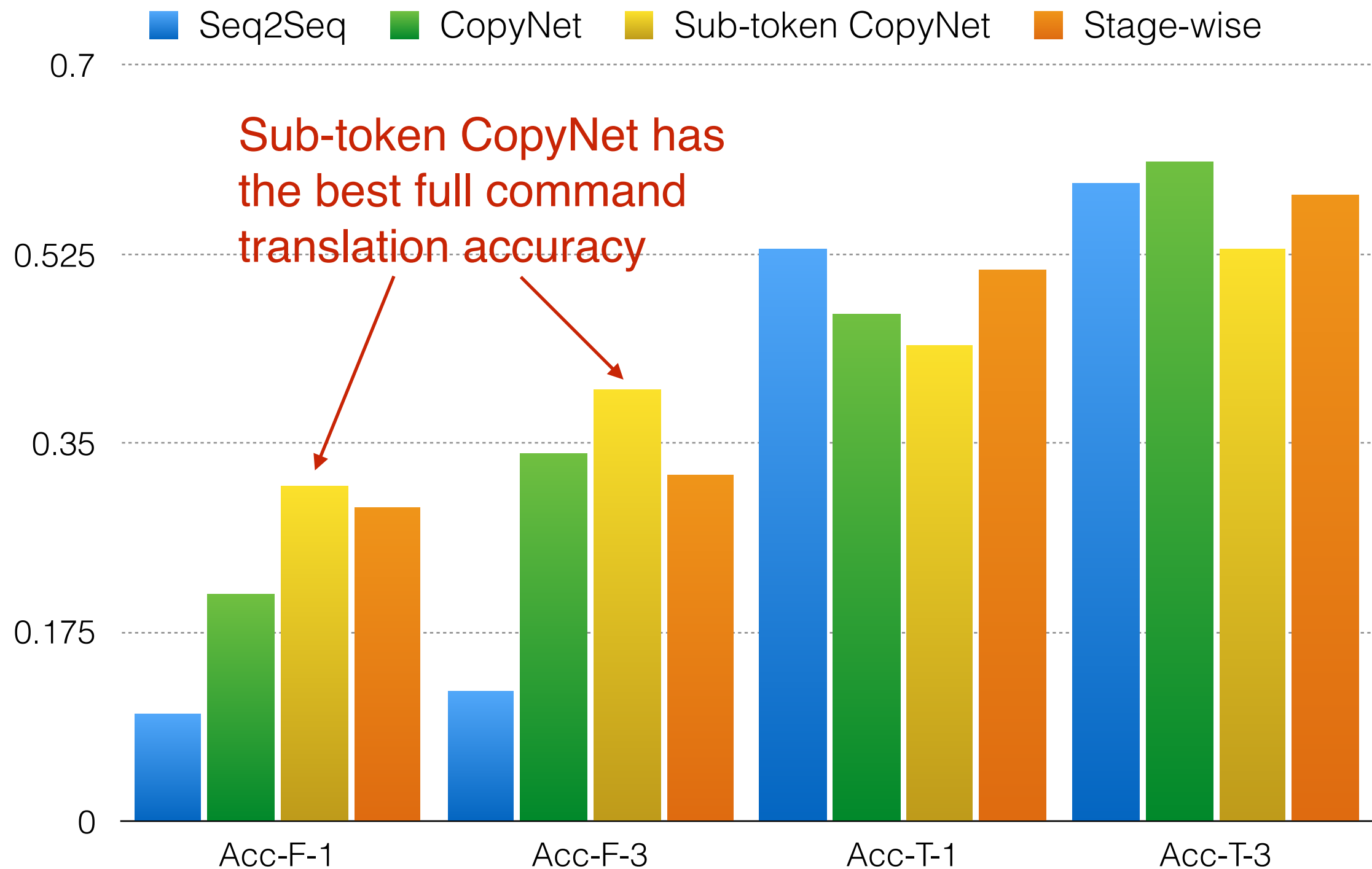
SYSTEM PERFORMANCE (Dev Set)



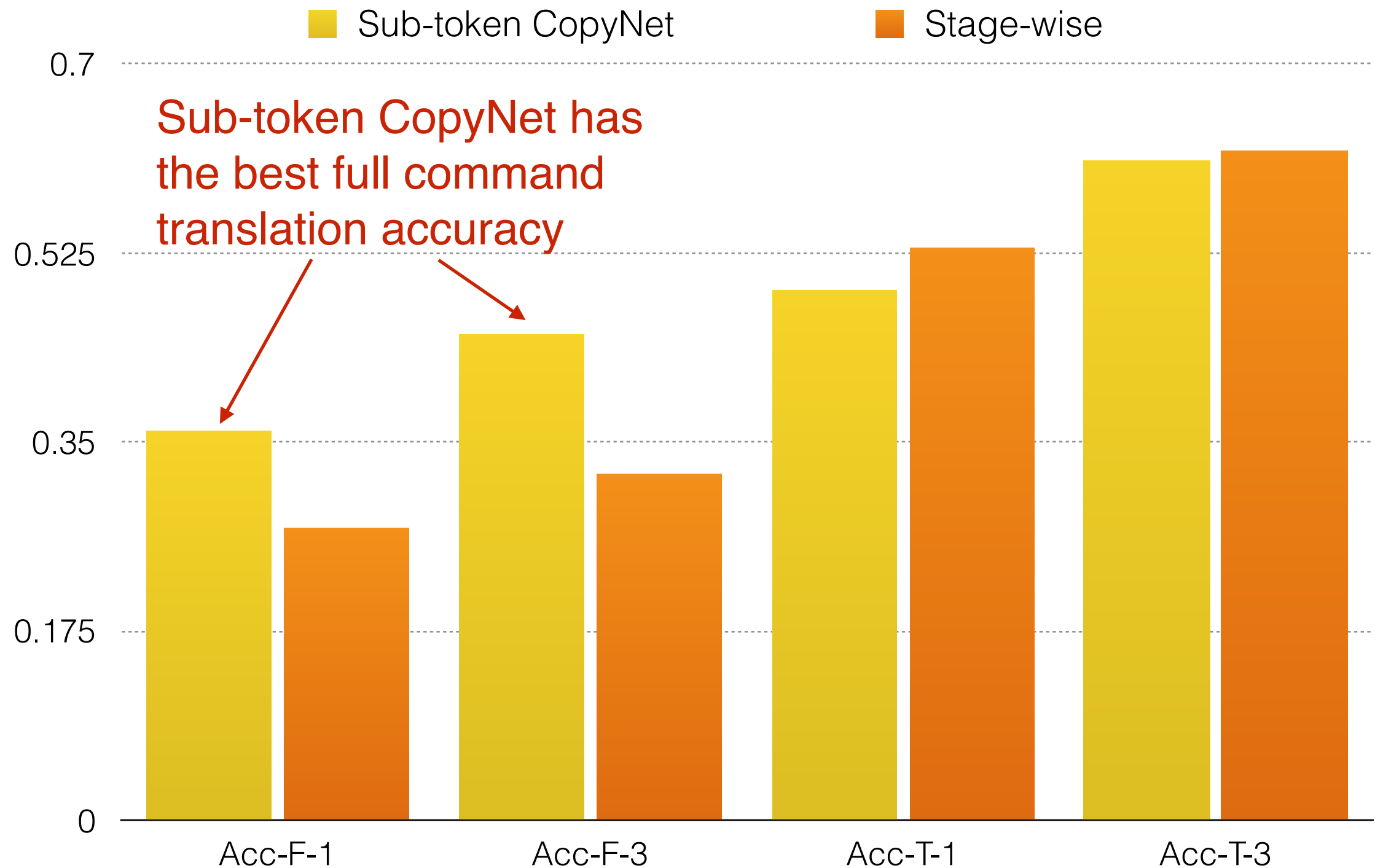
SYSTEM PERFORMANCE (Dev Set)



SYSTEM PERFORMANCE (Dev Set)



SYSTEM PERFORMANCE (Test Set)



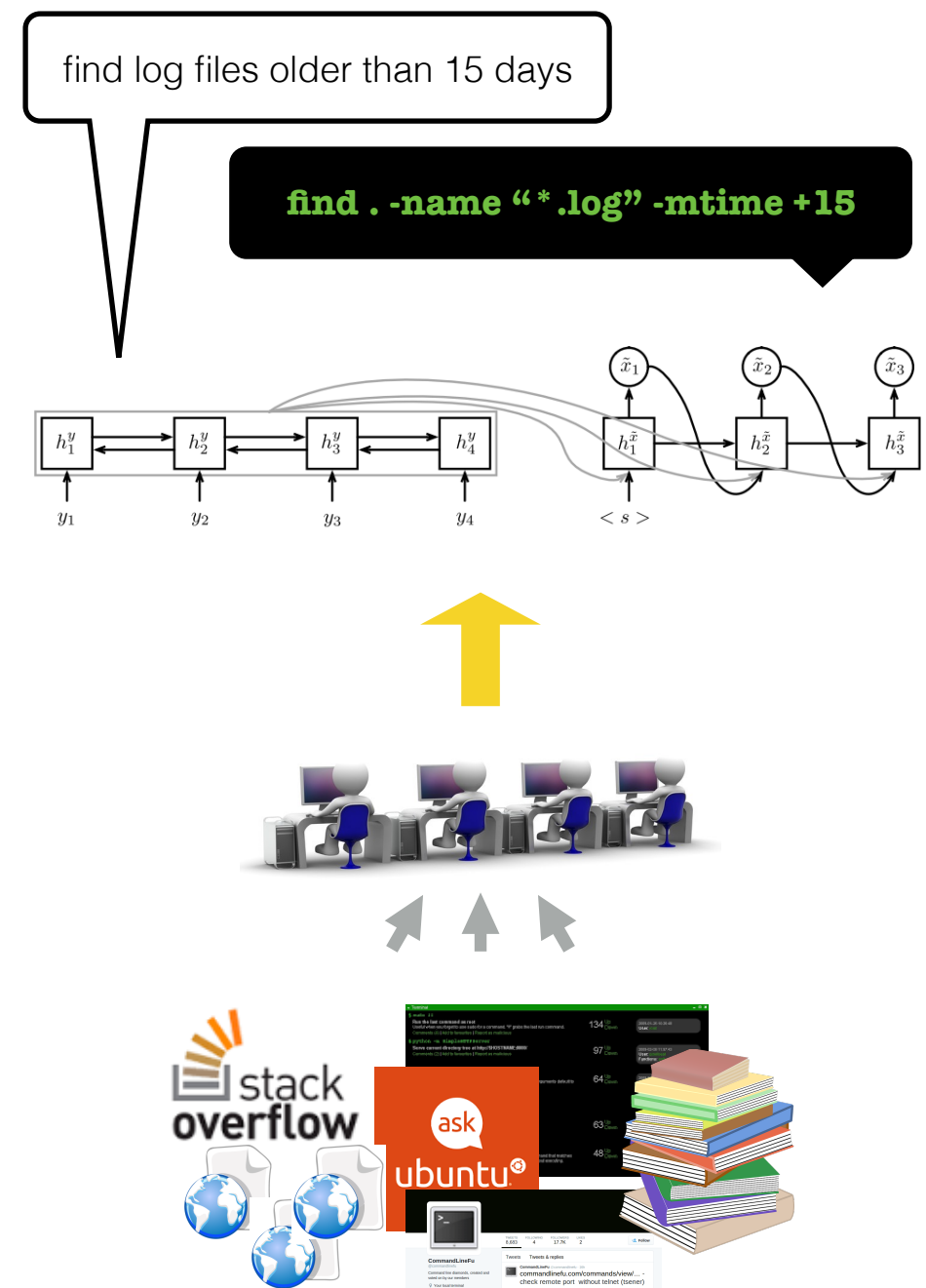
QUALITATIVE ANALYSIS

- Live Demo: <http://tellina.rocks>
- Split '/usr/bin/gcc' into 10 files of about equal size
- Which files in the computer were modified more than 30 days ago and larger than 500M
- Find all *company* (case-insensitive) files/directories under /basedir with null character as the delimiter

Github: <https://github.com/TellinaTool>

Demo: <http://tellina.rocks>

- **Corpus:** 10k real-world bash commands, paired with human-written English descriptions
- **Data-driven baselines:** motivated by SOTA neural machine translation approaches *copying, sub-token modeling*
- **Huge space for improvements**
- To appear in LREC 2018 conference proceedings
- Contact: xilin@salesforce.com

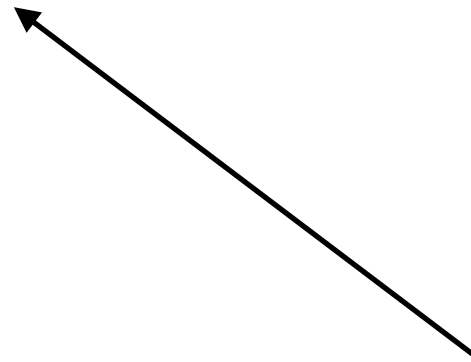


BKI - SEQ2SEQ OUTPUT PROBABILITY

Generation Probability



|target vocabulary|



BKII - COPYNET OUTPUT PROBABILITY

Generation Probability

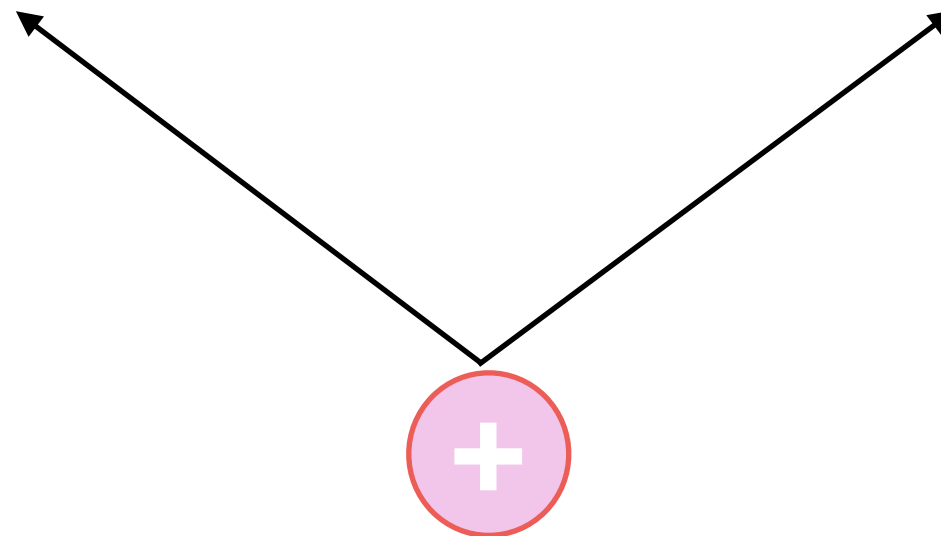


|target vocabulary|

Copy Probability



|source sequence|



BKIII - COPYNET (Gu et. al. 2016)

Generation Probability



|target vocabulary|

Copy Probability



|source sequence|

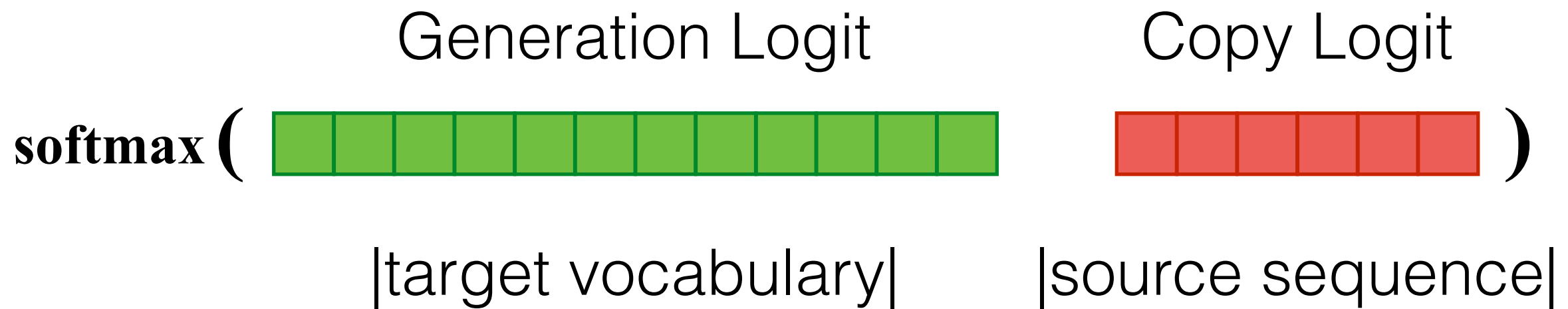
$$p(y_t | \mathbf{s}_t, y_{t-1}, \mathbf{c}_t, \mathbf{M}) = p(y_t, \mathbf{g} | \mathbf{s}_t, y_{t-1}, \mathbf{c}_t, \mathbf{M})$$

$$+ p(y_t, \mathbf{c} | \mathbf{s}_t, y_{t-1}, \mathbf{c}_t, \mathbf{M})$$

“hidden state”

“copying context”

BKIV- COPYNET (Gu et. al. 2016)



$$p(y_t, \mathbf{g}|\cdot) = \begin{cases} \frac{1}{Z} e^{\psi_g(y_t)}, & y_t \in \mathcal{V} \\ 0, & y_t \in \mathcal{X} \cap \bar{\mathcal{V}} \\ \frac{1}{Z} e^{\psi_g(\text{UNK})} & y_t \notin \mathcal{V} \cup \mathcal{X} \end{cases}$$

$$p(y_t, \mathbf{c}|\cdot) = \begin{cases} \frac{1}{Z} \sum_{j: x_j = y_t} e^{\psi_c(x_j)}, & y_t \in \mathcal{X} \\ 0 & \text{otherwise} \end{cases}$$

BKV - SPEED-UP EXPERT SOURCING

Command2NL Logout (victoria-lin)

mkdir join set source touch env ln uname which cd

chown mount tee more hostname split mktemp od column file

read ssh yes basename less nl rsync zcat rev readlink

shopt paste who fold gzip seq tr whoami comm scp

su tree mv tac jobs pwd ssh-keygen gunzip alias ⁵²head

cat cpio date dig export chmod dirname history kill ping

sleep top crontab md5sum rmdir awk cut tail cal rename

df diff rm watch ls md5 uniq curl screen ps

chgrp pstree cp nohup sort w bind tar wget apt-get

awk

urls annotated: 21

pairs annotated: 356

Figure 2. Data Collection Interface Screenshot

BKVI - THREE-STAGE TRANSLATION APPROACH

natural language input:

find all log files older than 15 days



**Stage 1: rule-based open-vocabulary
entity recognition**

entity mentions: {filename: “log”,
timespan: “15 days”}

natural language template:

find all [filename] files older than [timespan]

**Stage 3: Argument filling and
post-processing**

synthesized program templates:

find . -name “*.log” -mtime
+15d



find . -type f -name “*.log” -mtime
+15d

...

**Stage 2: NL template to program
template translation**