

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
```

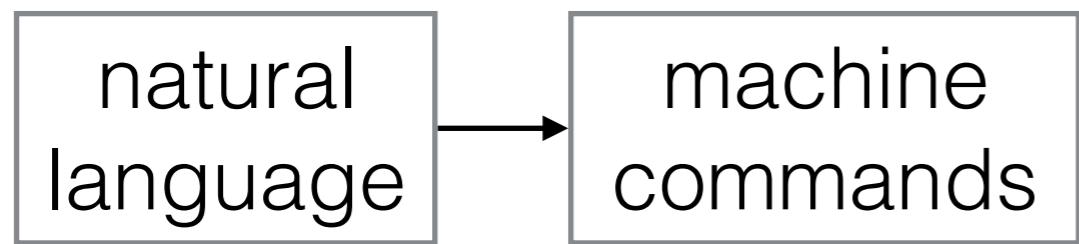
Victoria Lin[↑] Chenglong Wang[↑] Luke Zettlemoyer[↑]

Michael D. Ernst[↑]

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

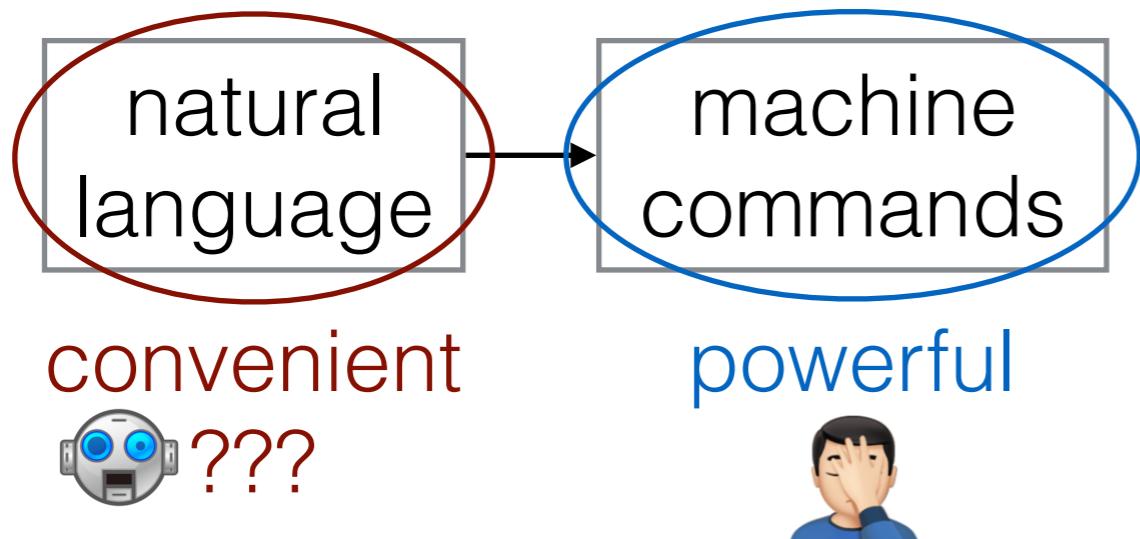
OUTLINE

Problem Definition



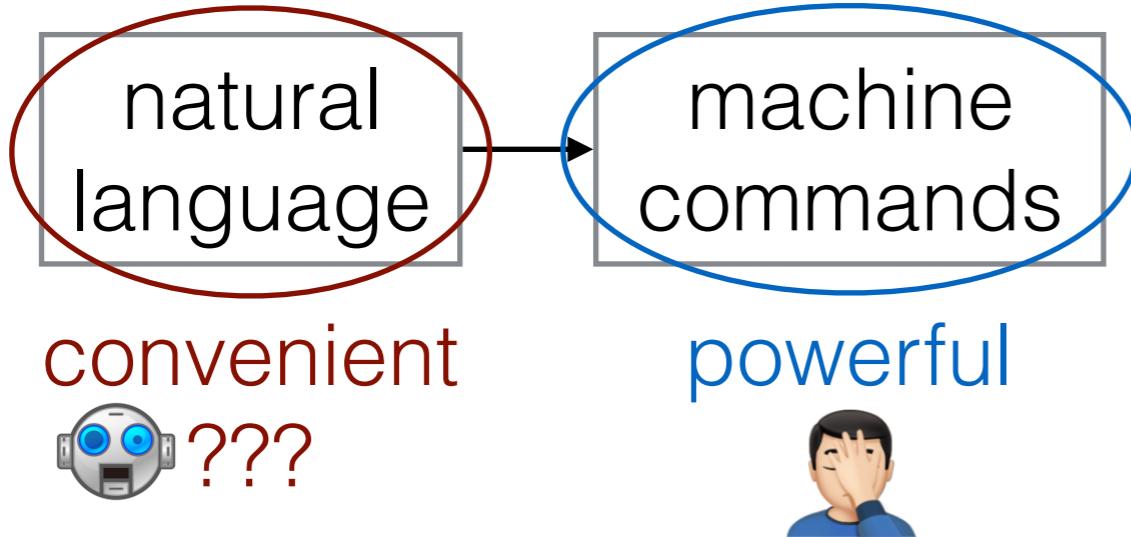
OUTLINE

Problem Definition



OUTLINE

Problem Definition



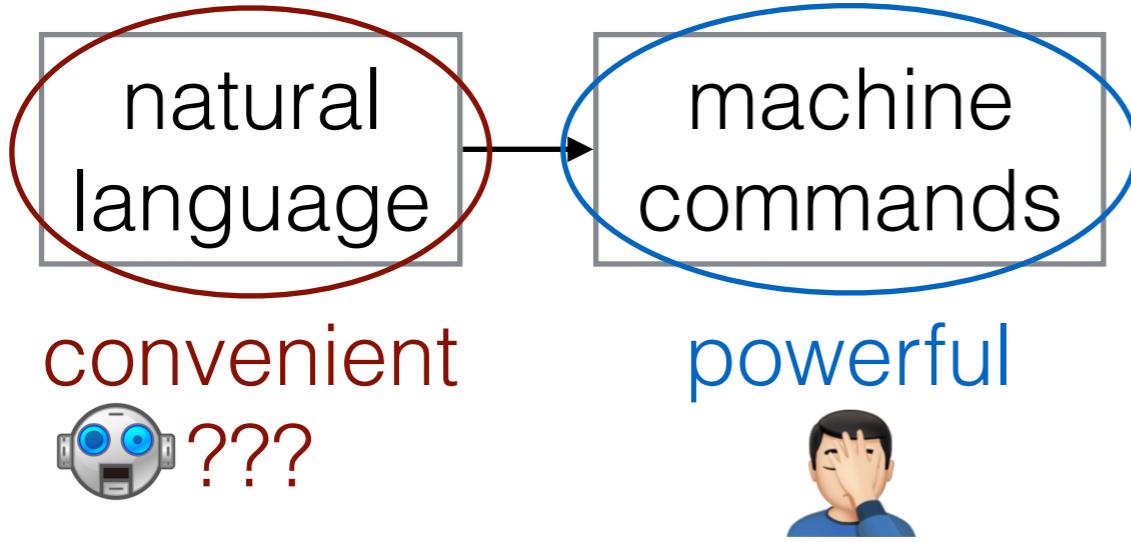
Domain

A screenshot of a terminal window titled "1. bash". The window displays a file listing and several command inputs:

```
total 0
drex--xr-x 22 xilinx staff 748 Jun 9 2016 helper
drex--xr-x 9 xilinx staff 306 Apr 21 2016 opoleye_
drex--xr-x 7 xilinx staff 238 Jun 9 2016 opoleye_kbp
drex--xr-x 5 xilinx staff 170 Dec 17 2015 pigwidgen
drex--xr-x 13 xilinx staff 442 Mar 15 22:47 reflexnet
drex--xr-x 25 xilinx staff 50 Mar 15 22:47 reflexnet_
drex--xr-x 26 xilinx staff 884 Dec 18 2015 skin
drex--xr-x 22 xilinx staff 748 Jun 9 2016 topicoa
drex--xr-x 31 xilinx staff 1054 Feb 21 14:13 task_platform
drex--xr-x 16 xilinx staff 544 Mar 29 18:18 telling
drex--xr-x 50 xilinx staff 1704 Mar 29 18:18 telling_Fee
drex--xr-x 51 xilinx staff 574 Mar 31 21:02 topicoa_github.io
drex--xr-x 62 xilinx staff 2108 Oct 9 2015 tutor
drex--> 16 xilinx staff 544 Feb 22 12:17 type-cog
VictorLos-MacBook-Pro-2:Projects xilinx$ find . -name beam_search.py
./beam_search.py
./reflexnet/Net/beam_search.py
./telling/telling_beam_search.py
VictorLos-MacBook-Pro-2:Projects xilinx$ find . -name "beam_search.py" -mtime -1
./helper/encoder_decoder/beam_search.py
VictorLos-MacBook-Pro-2:Projects xilinx$ find all files named "beam_search.py"
that was modified today
```

OUTLINE

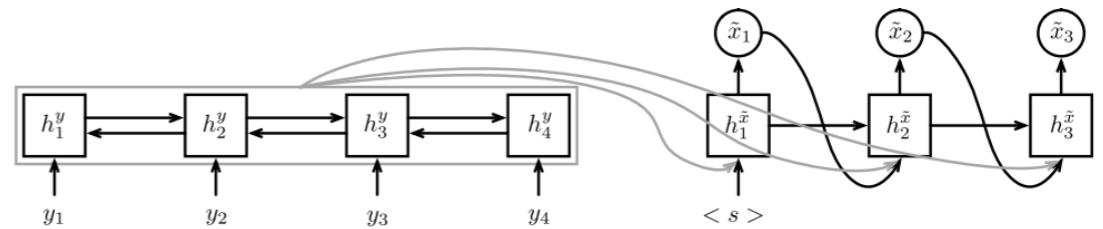
Problem Definition



Domain

```
total 0
drwxr-xr-x 22 xilinx staff 748 Jun 9 2016 helper
drwxr-xr-x 9 xilinx staff 306 Apr 21 2016 oppleye_kbp
drwxr-xr-x 7 xilinx staff 238 Jun 9 2016 oppleye_kbp
drwxr-xr-x 5 xilinx staff 170 Dec 17 2015 pigwidgen
drwxr-xr-x 13 xilinx staff 442 Mar 15 22:47 reflexnet
drwxr-xr-x 2 xilinx staff 64 Feb 21 14:13 setup
drwxr-xr-x 26 xilinx staff 884 Dec 18 2015 skin
drwxr-xr-x 22 xilinx staff 748 Jun 9 2016 topicoa
drwxr-xr-x 31 xilinx staff 1054 Feb 21 14:13 task_platform
drwxr-xr-x 16 xilinx staff 544 Mar 29 18:18 telling
drwxr-xr-x 50 xilinx staff 1708 Mar 29 18:18 telling_Fee
drwxr-xr-x 11 xilinx staff 374 Mar 31 21:02 tensorflow_github.io
drwxr-xr-x 62 xilinx staff 2188 Oct 9 2015 tutor
drwxr-xr-x 16 xilinx staff 544 Feb 12 12:17 type-cog
Victorios-MacBook-Pro-2:Projects xilinx [1]: ls -l display all files
Victorios-MacBook-Pro-2:Projects xilinx [2]: find . -name "beam_search.py"
./beam_search.py
./reflexnet/Net/beam_search.py
./telling/telling_beam_search.py
Victorios-MacBook-Pro-2:Projects xilinx [3]: find . -name "beam_search.py" -mtime -1
./helper/encoder_decoder/beam_search.py
Victorios-MacBook-Pro-2:Projects xilinx [4]: find all files named "beam_search.py"
that was modified today
```

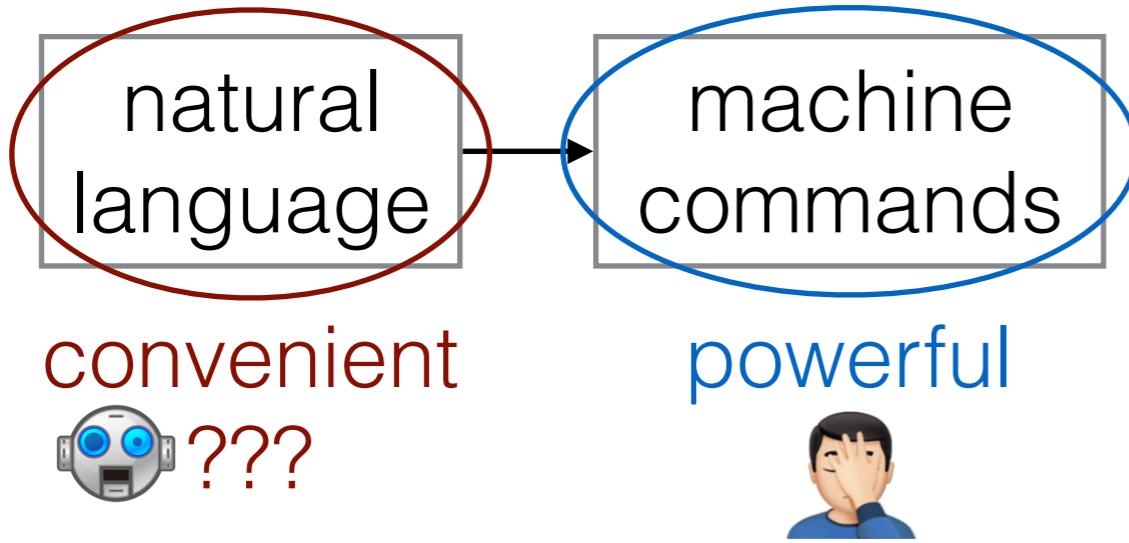
Data-Driven Approaches



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

OUTLINE

Problem Definition



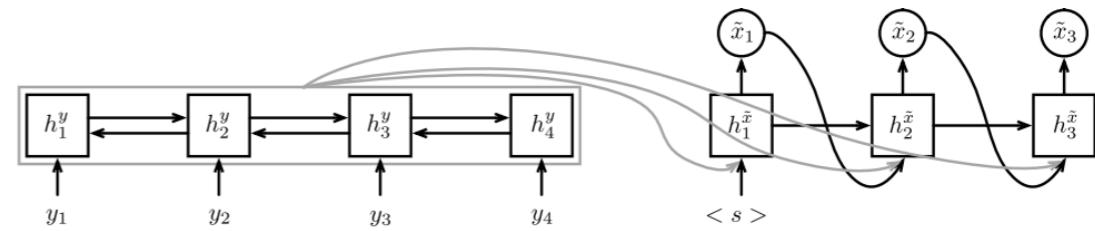
Corpus Construction



Domain

A screenshot of a terminal window titled '1. bash' showing a file search process. The command 'find . -name "beam_search.py"' is run, and the output lists several files named 'beam_search.py' across different paths, such as 'beam_search.py' in 'beam_search.py' and 'beam_search.py' in 'beam_search.py'.

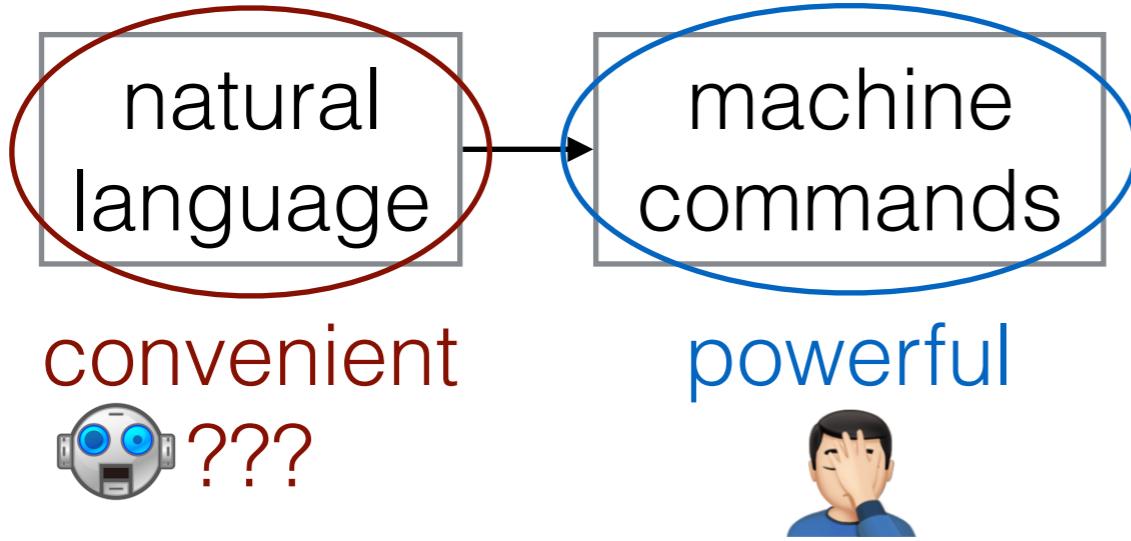
Data-Driven Approaches



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

OUTLINE

Problem Definition



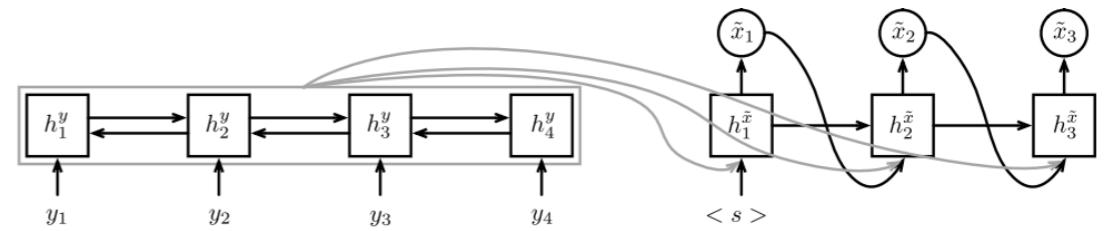
Corpus Construction



Domain

A screenshot of a terminal window titled "1. bash". The window displays a series of command-line inputs and their outputs. The user runs "ls -l" to show all files, then "find . -name beam_search.py" to search for files named "beam_search.py", and finally "find . -name beam_search.py -mtime -1" to find files modified today. The terminal interface includes a green selection bar at the top and several command-line history entries.

Data-Driven Approaches



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

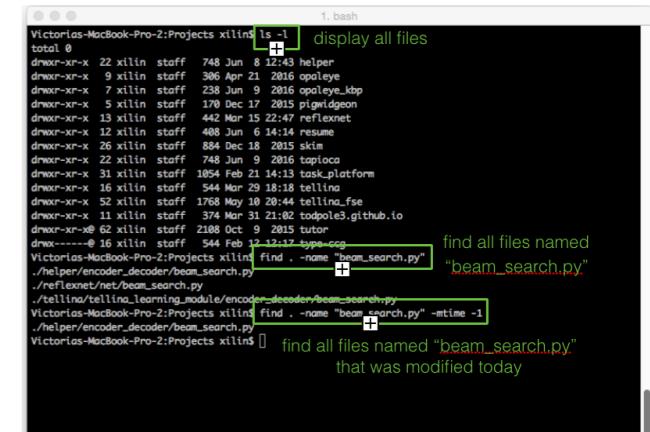
System Performance Qualitative Analysis Live Demo

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



A screenshot of a macOS terminal window titled "1.bash". The window shows a command history of bash commands entered by the user. The commands include file listing (ls -l), displaying all files (ls -l), finding files named "beam_search.py" (find . -name "beam_search.py"), and finding files modified today (find . -mtime -1). The terminal window has a dark theme.

```
Victorios-MacBook-Pro-2:Projects xilin$ ls -l display all files
total 0
Victorios-MacBook-Pro-2:Projects xilin$ ./beam_search.py
find all files named "beam_search.py"
Victorios-MacBook-Pro-2:Projects xilin$ ./beam_search.py
find all files named "beam_search.py"
Victorios-MacBook-Pro-2:Projects xilin$ ./beam_search.py
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

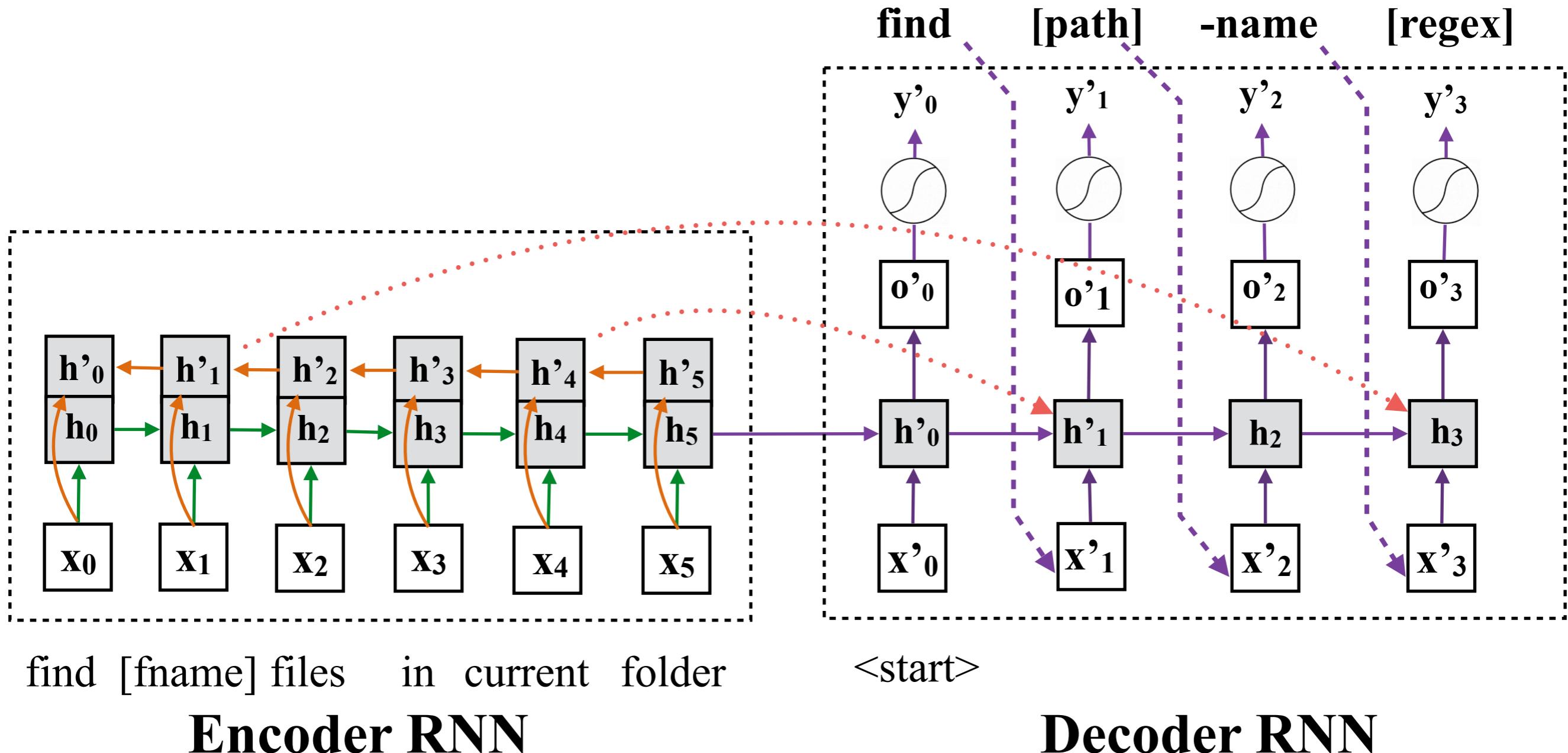
- Neural Networks: Natural Language → Formal Languages

- ✓ 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

Incorporating Copying Mechanism in Sequence-to-Sequence Learning, Gu et. al. EMNLP 2016

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

Incorporating Copying Mechanism in Sequence-to-Sequence Learning, Gu et. al. EMNLP 2016

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™**
- Collect bash commands and their natural language descriptions from the web



- ✓ web interface to control the collection process

BASH COMMAND FILTERING

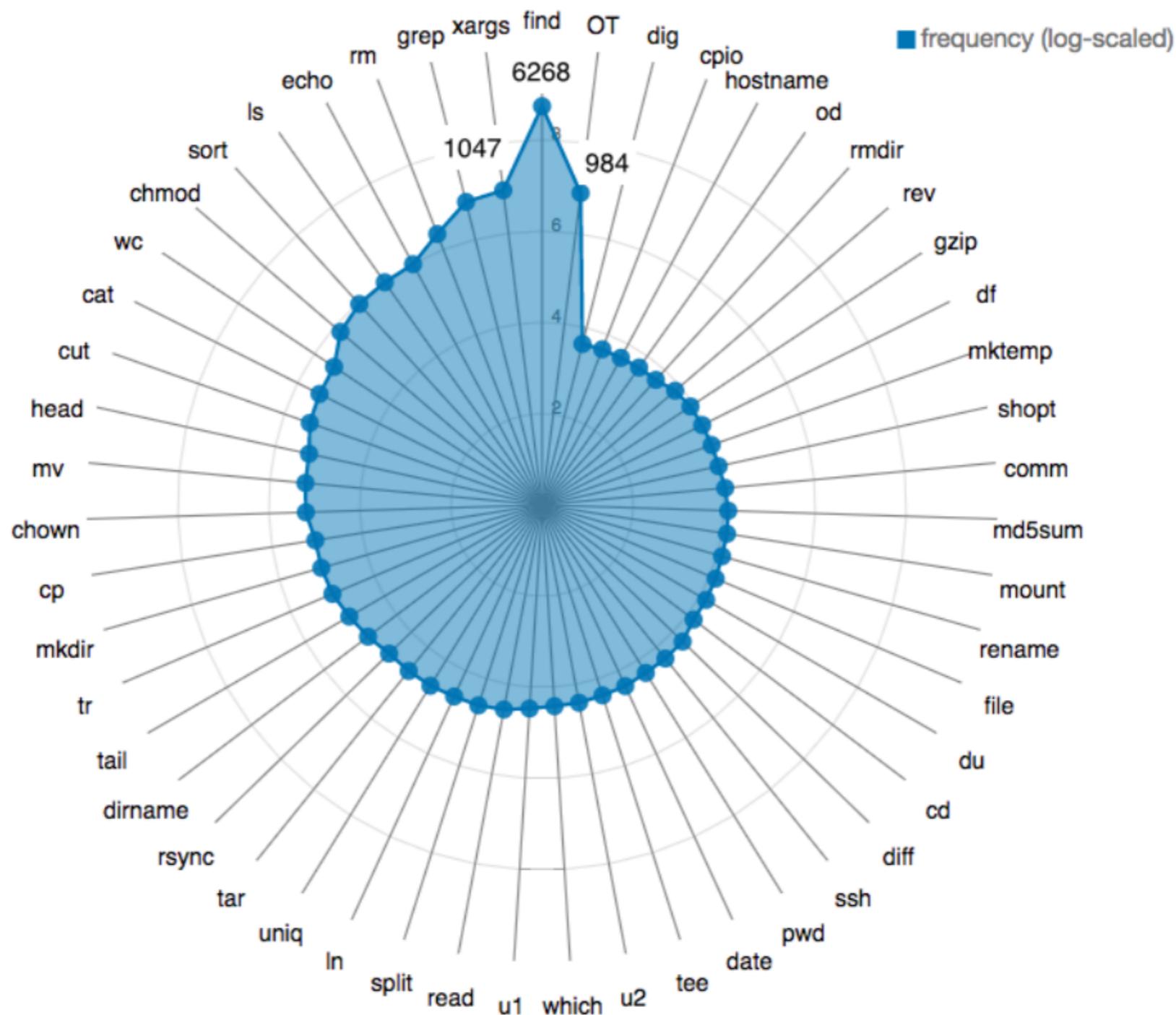
- Bash Command

In-scope	Single command	
	Logical connectives	&&, , ()
	Nested command	pipeline 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.

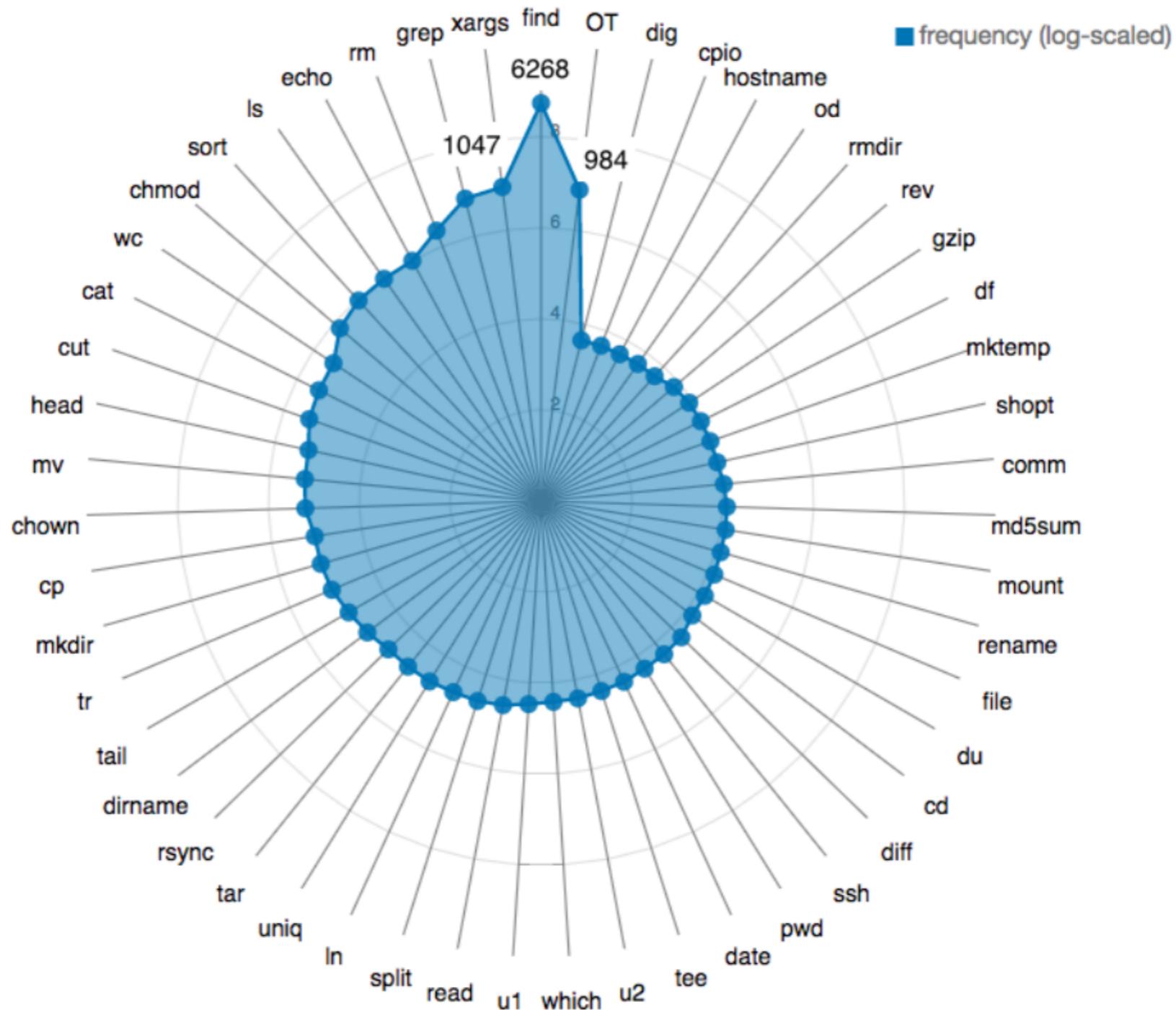
DATA STATISTICS

- 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

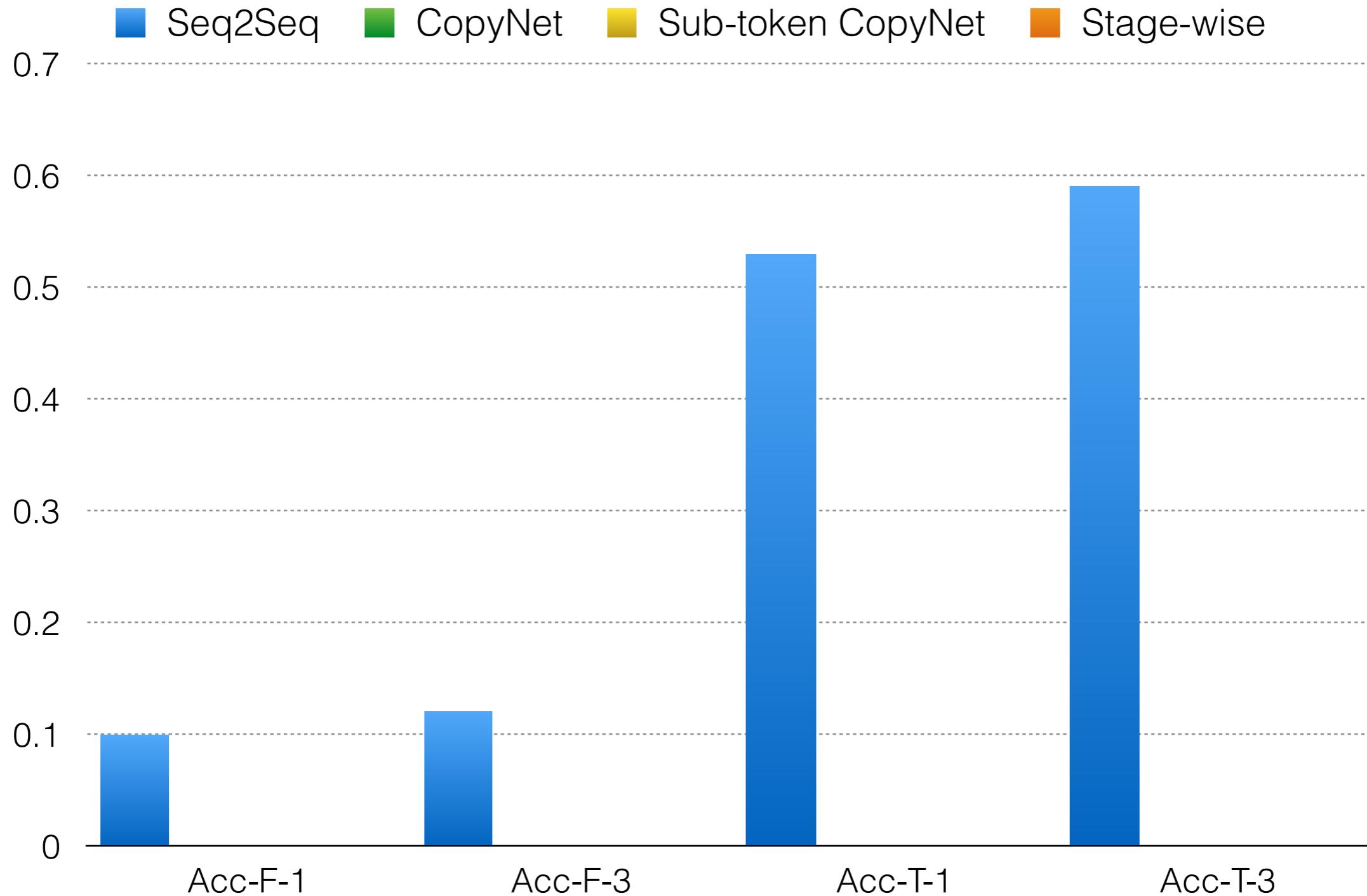
EVALUATION METHODOLOGY

- 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

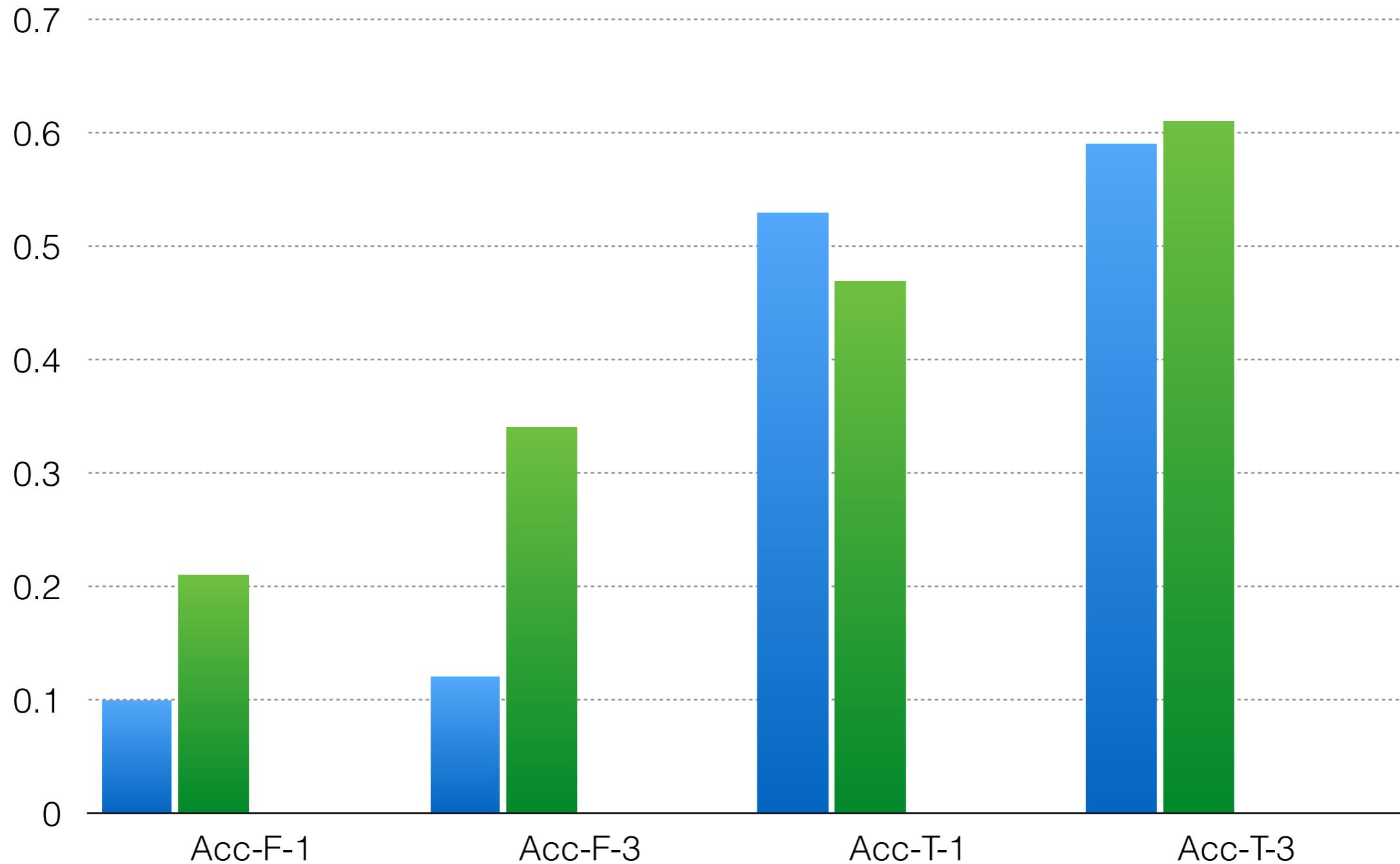
- Vanilla Seq2Seq (Sutskever et. al. '14)
- CopyNet (Gu et. al. '17)
- Three-stage translation model (Lin et. al. '17)
 1. Convert both NL and bash command to templates
 2. Apply Seq2Seq translation on the templates
 3. Fill arguments using heuristics

SYSTEM PERFORMANCE (Dev Set)

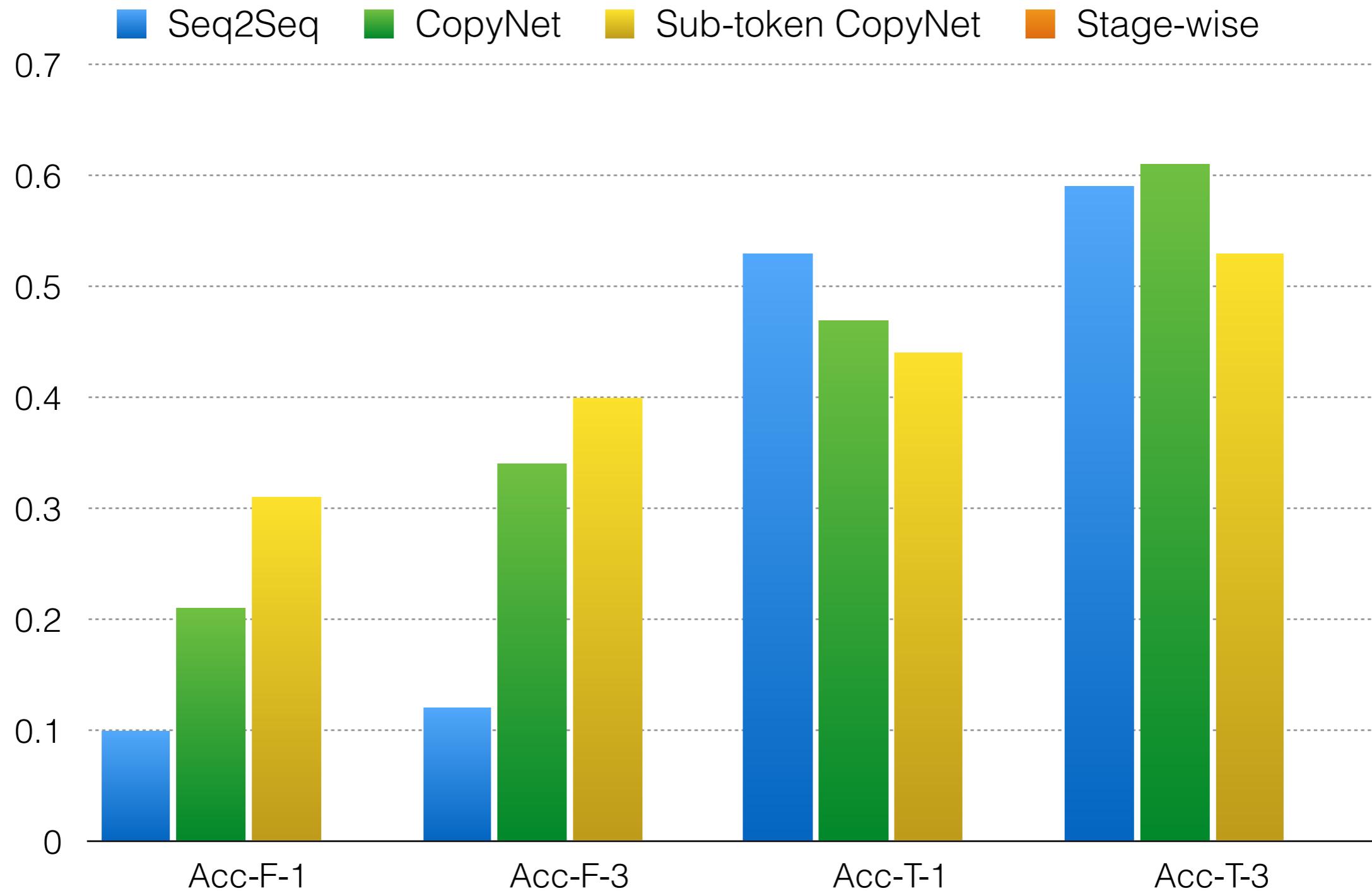


SYSTEM PERFORMANCE (Dev Set)

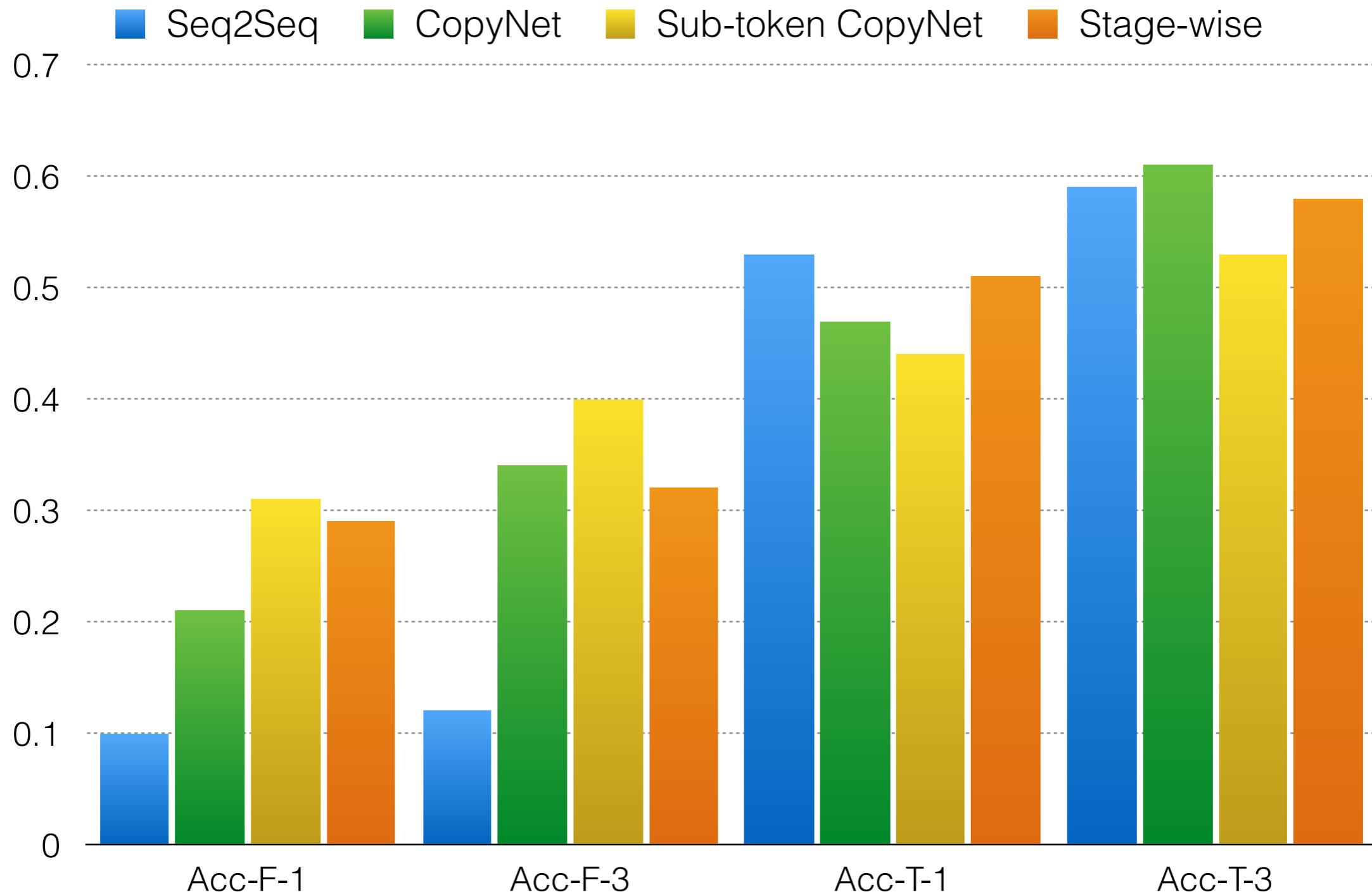
Seq2Seq CopyNet Sub-token CopyNet Stage-wise



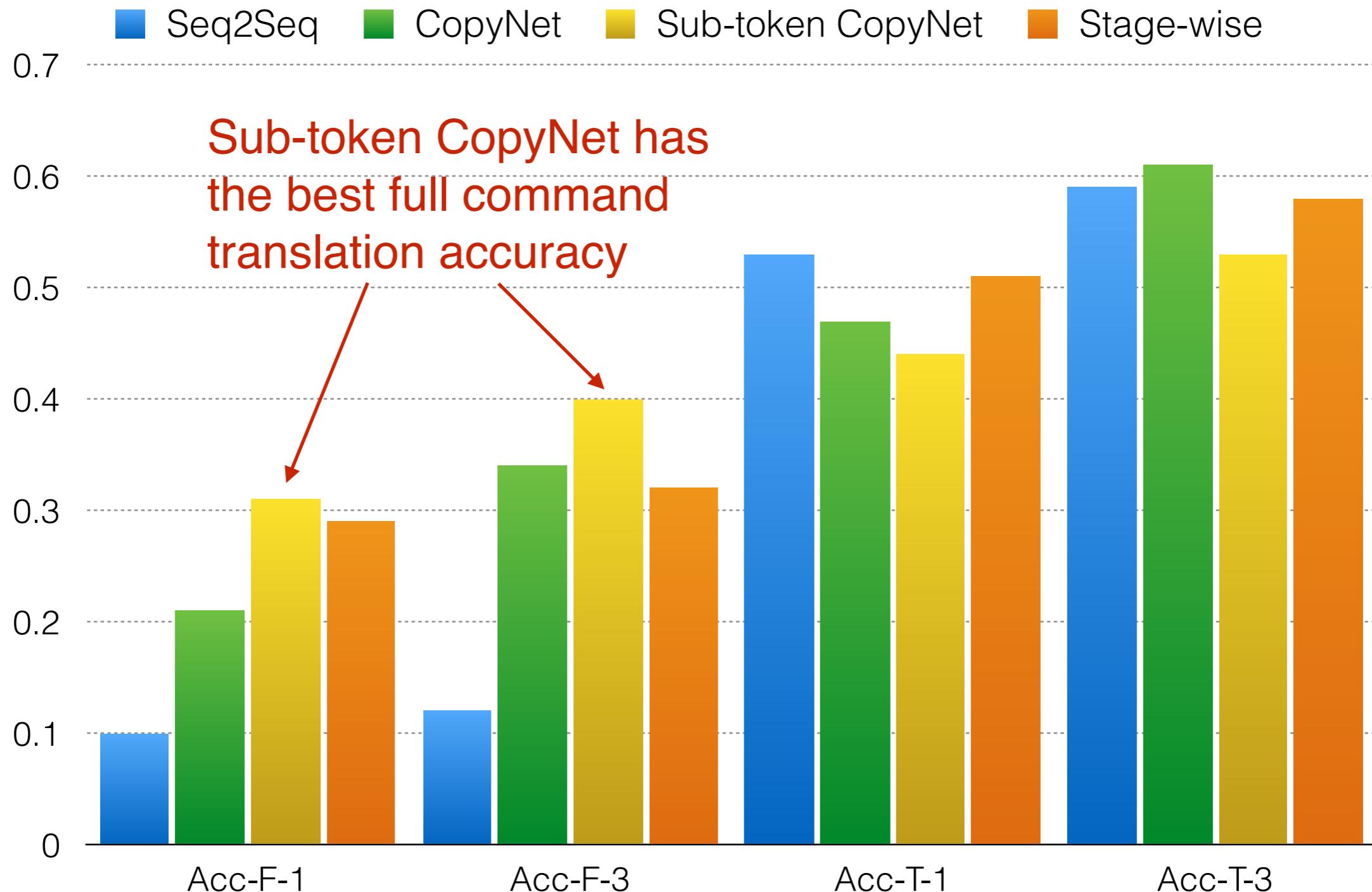
SYSTEM PERFORMANCE (Dev Set)



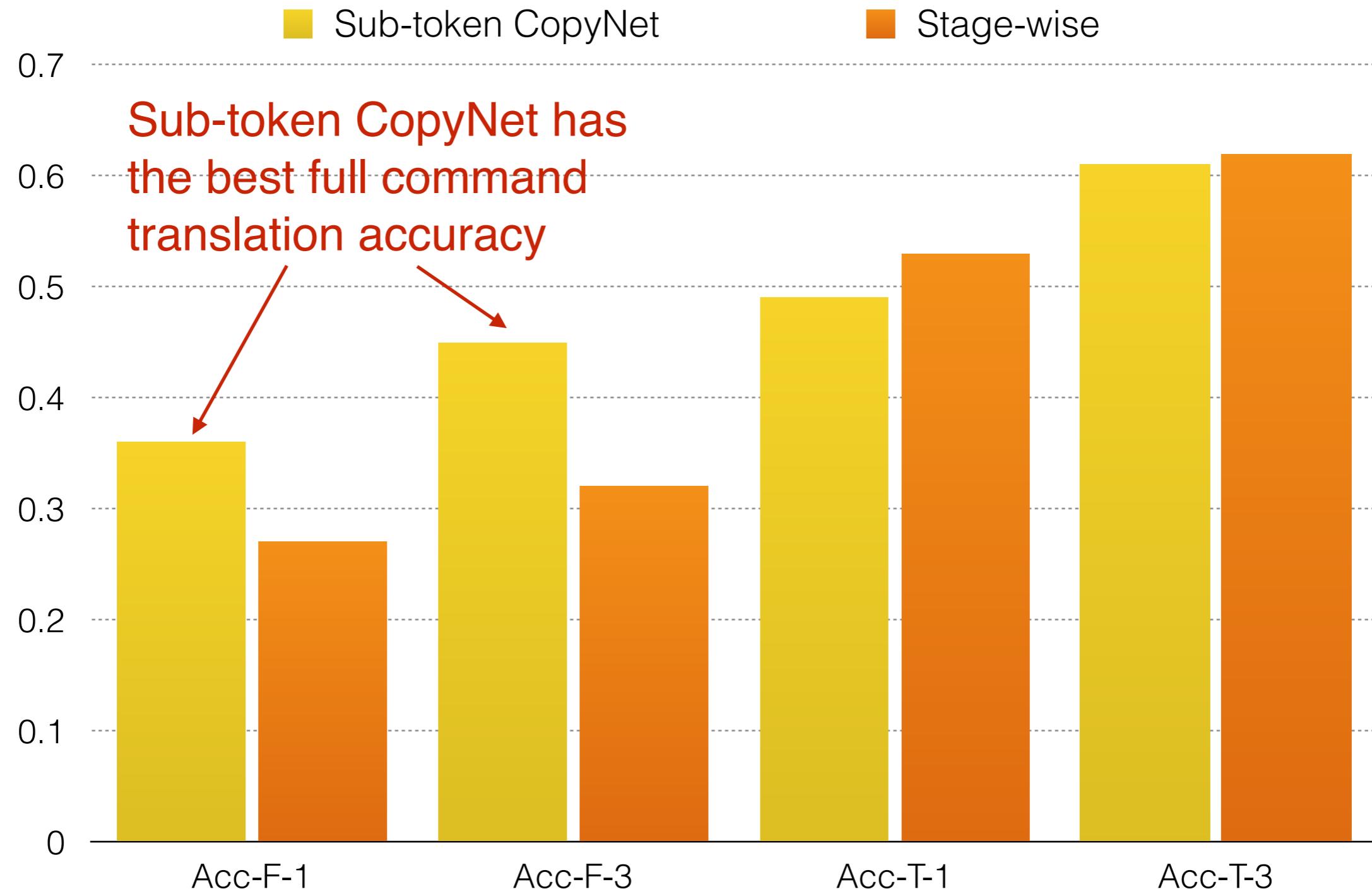
SYSTEM PERFORMANCE (Dev Set)



SYSTEM PERFORMANCE (Dev Set)



SYSTEM PERFORMANCE (Test Set)



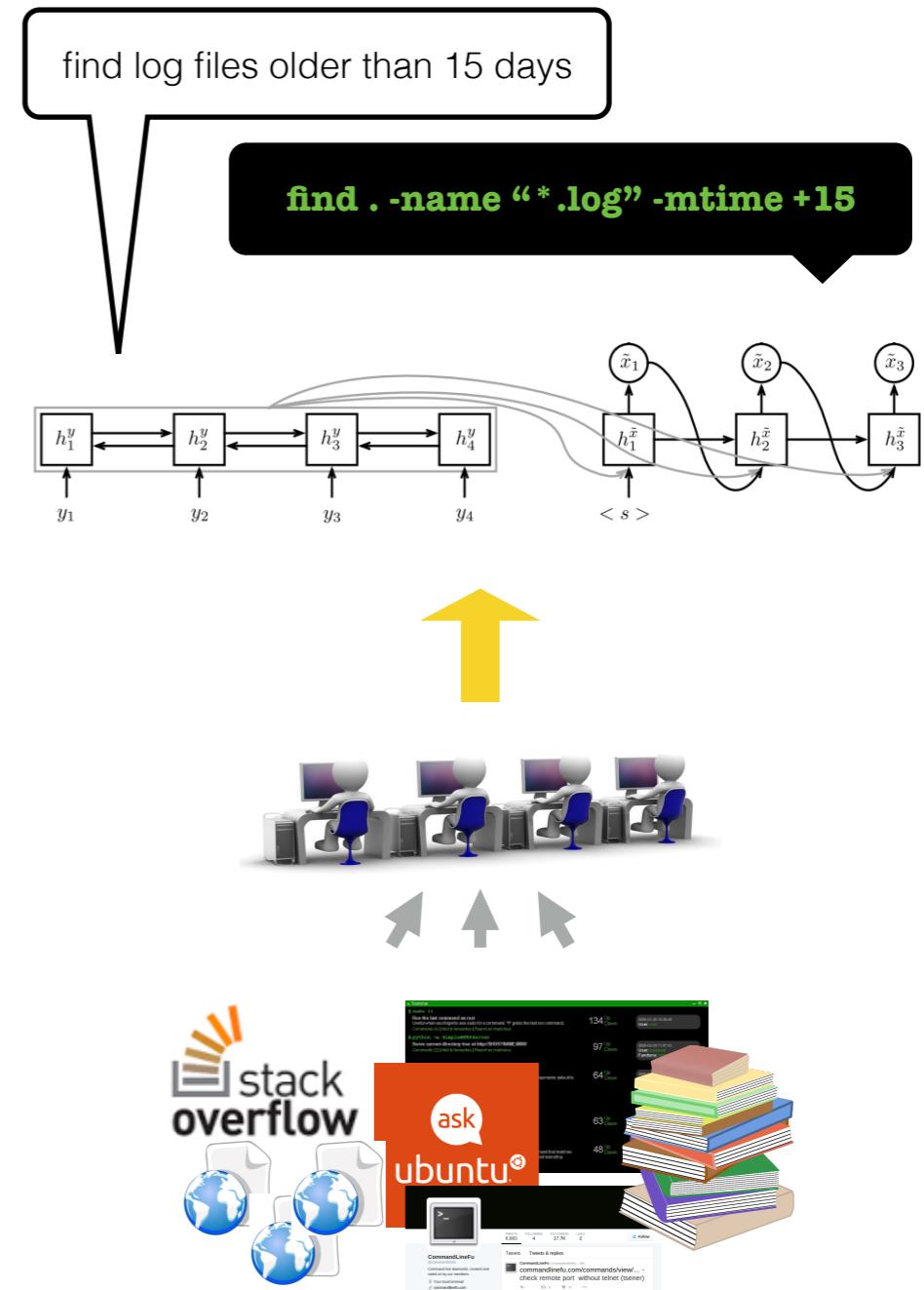
QUALITATIVE ANALYSIS

- 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

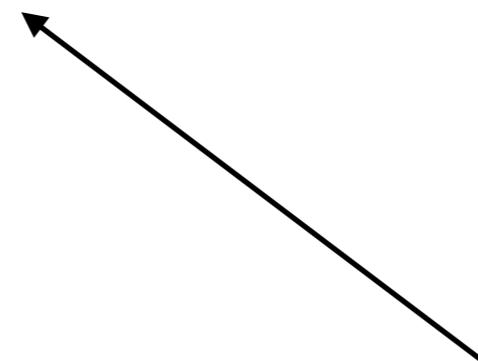


BKI - SEQ2SEQ OUTPUT PROBABILITY

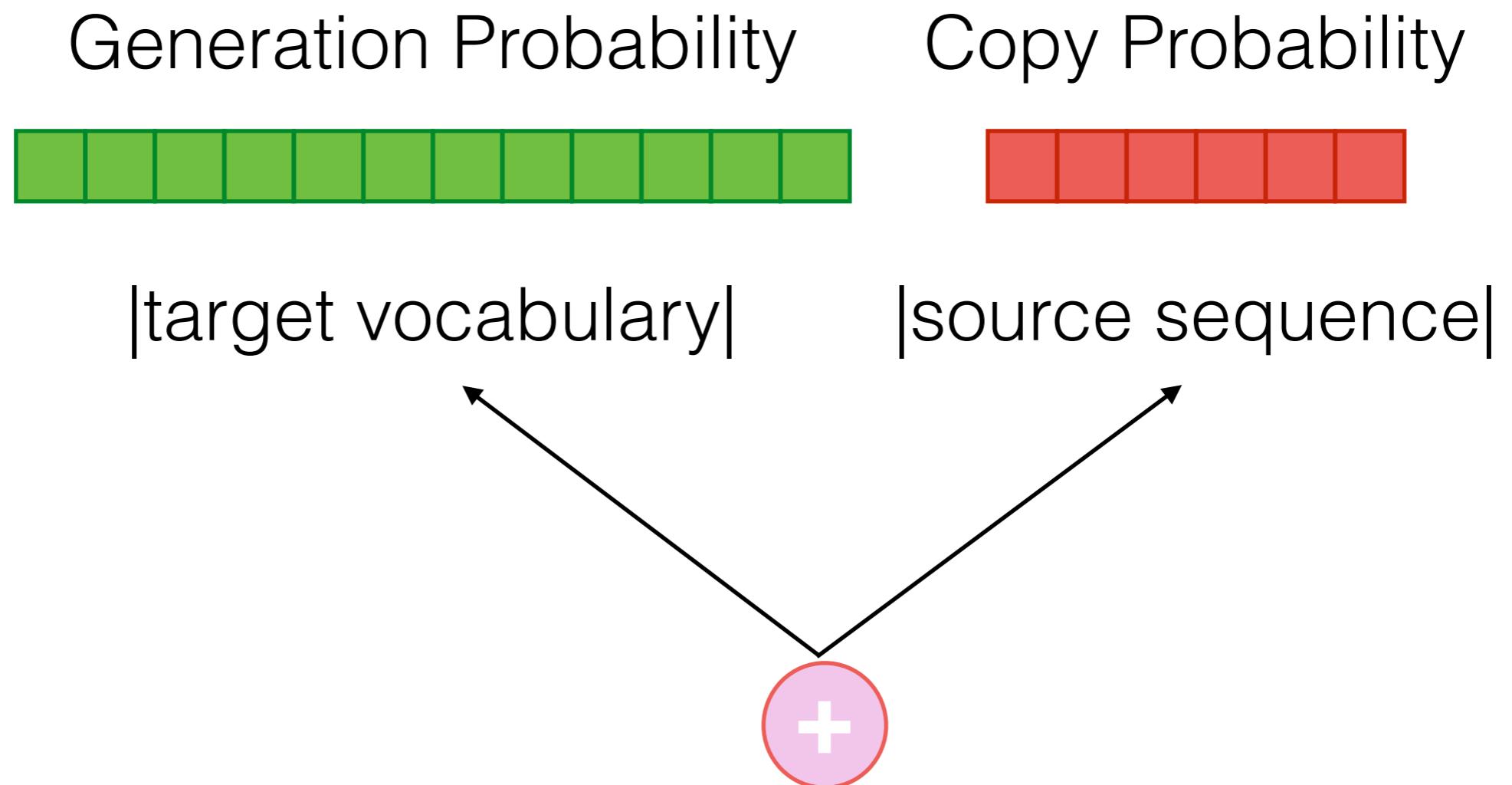
Generation Probability



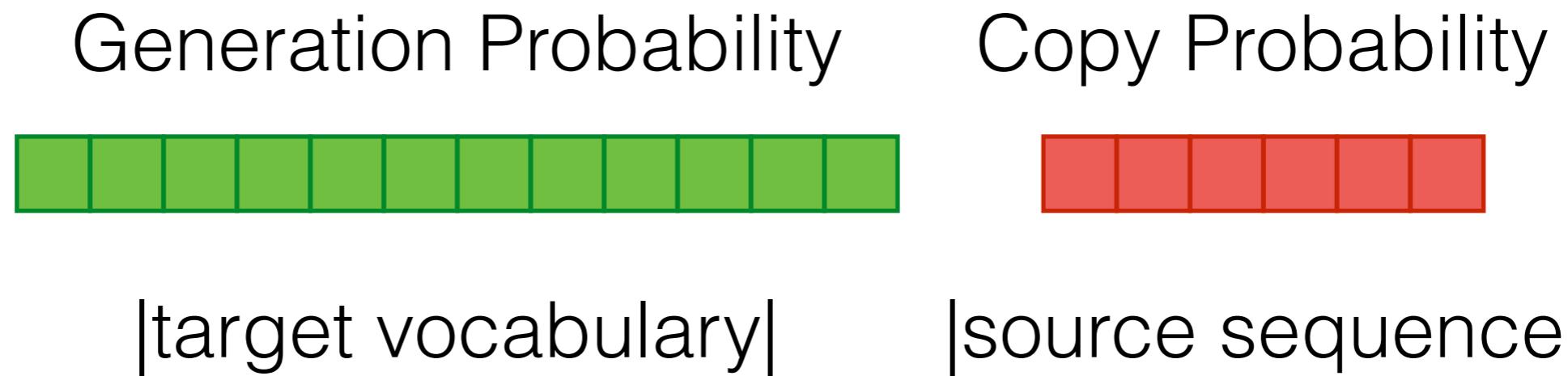
$|\text{target vocabulary}|$



BKII - COPYNET OUTPUT PROBABILITY



BKIII - COPYNET (Gu et. al. 2016)

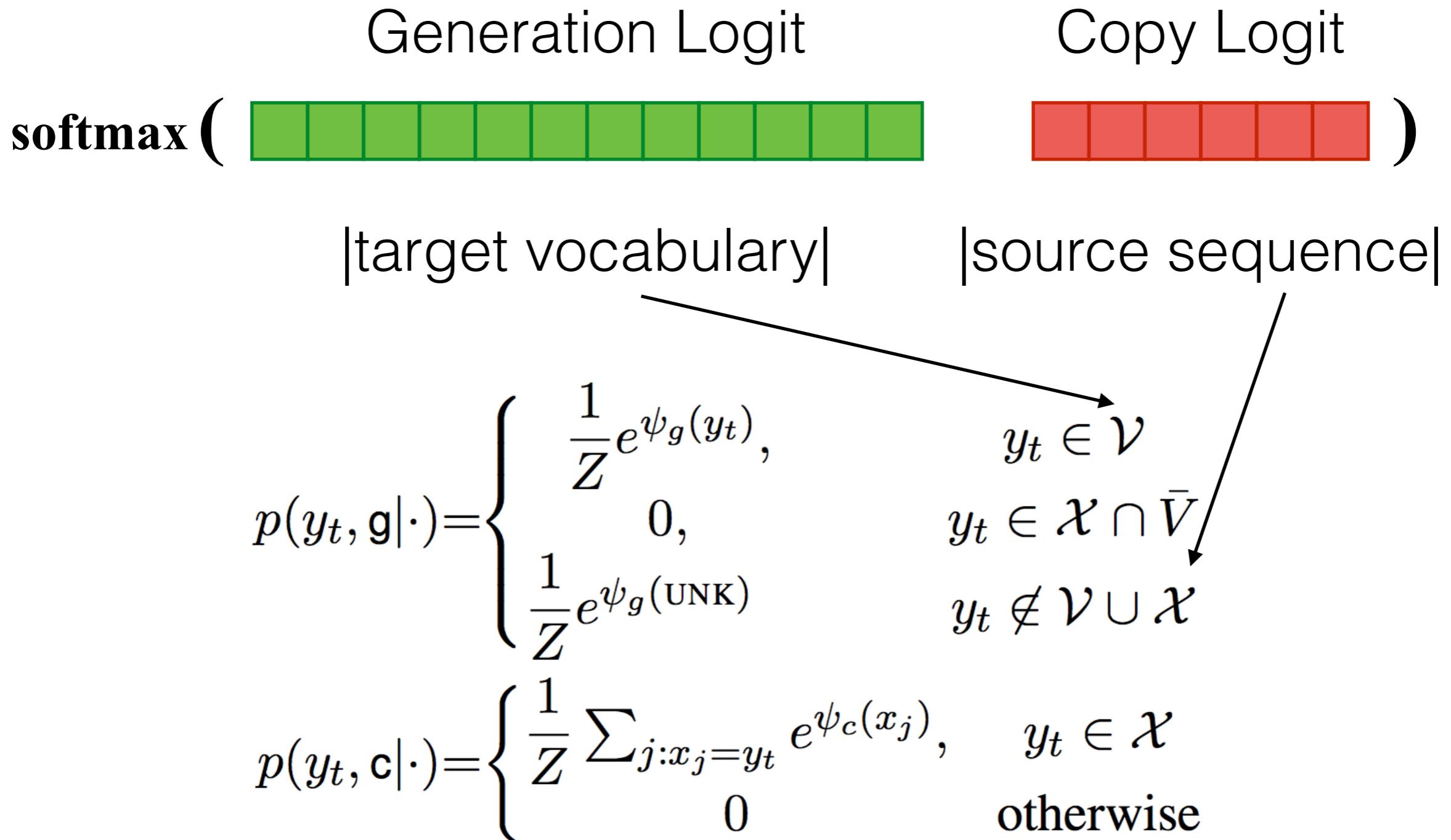


$$p(y_t | \mathbf{s}_t, y_{t-1}, \mathbf{c}_t, \mathbf{M}) = p(y_t, g | \mathbf{s}_t, y_{t-1}, \mathbf{c}_t, \mathbf{M})$$

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

↑ ↑
“hidden state” “copying context”

BKIV- COPYNET (Gu et. al. 2016)



BKV - SPEED-UP EXPERT SOURCING

Command2NL

Logout (victoria-lin)

mkdir join set source touch env ln uname which cd

awk

chown mount tee more hostname split mktemp od column file

urls annotated: 21

pairs annotated: 356

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 head

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

Figure 2. Data Collection Interface Screenshot

BKVI - THREE-STAGE TRANSLATION APPROACH

natural language input:

find all log files older than 15 days



Stage 3: Argument filling and post-processing

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]

synthesized program templates:

find . -name "*.log" -mtime
+15d

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

...

Stage 2: NL template to program template translation