THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Towards a Command-Line Autocorrect Tool

Grady McPeak
CSCI 6907 Final Project

Why Aren’t CLIs Fancier?

- Accessibility in CLIs
  - Lowering the barrier to becoming a “power user”
  - Increasing computer literacy of average users
  - GUIs are slow and heavy!
    - If this is going into a CLI, it needs to be as light as possible

Why Aren’t CLIs Fancier?

Can We Generate Typos?

- Command dataset from Ubuntu `man` pages
  - command names and associated flags

Provided by: coreutils_8.30-3ubuntu2_amd64

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILEs (the current directory by default). Sort entries alphabetically if none of **-cftuvSUX** nor **--sort** is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all
do not ignore entries starting with .

-A, --almost-all
do not list implied . and ..

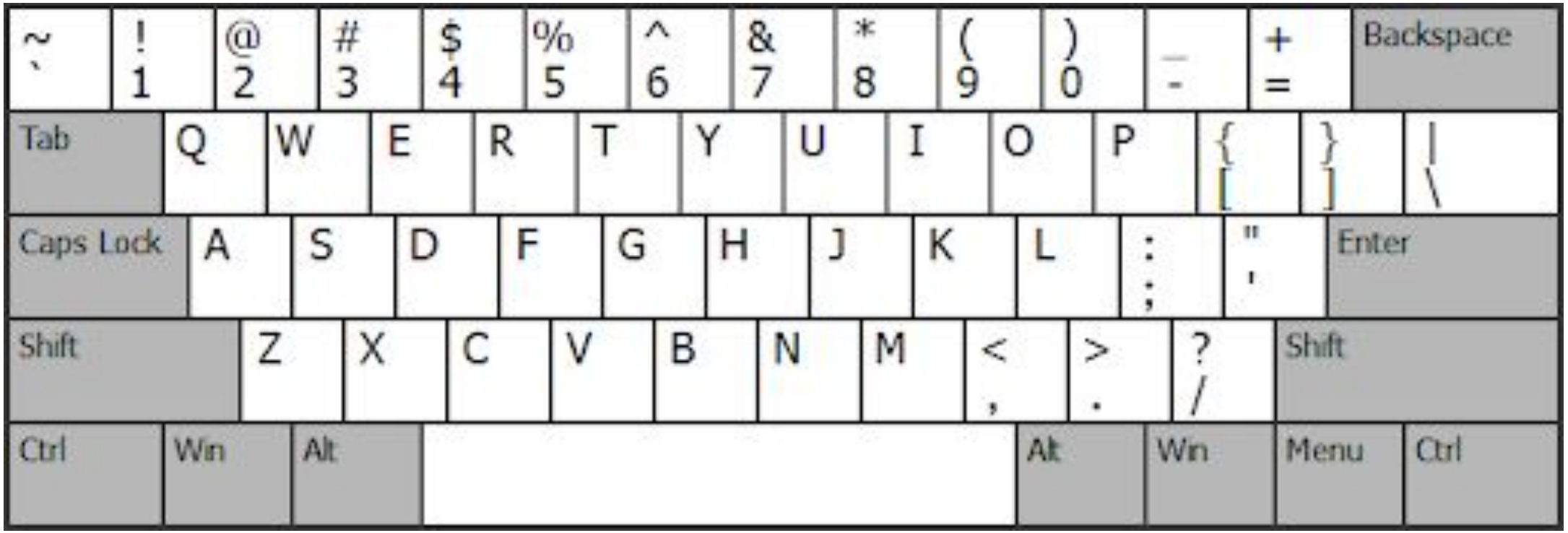
--author
with **-l**, print the author of each file

-b, --escape
print C-style escapes for nongraphic characters

Can We Generate Typos?

- *Synthetic dataset* of typos with associated flags
  - 1 or two incorrect keystrokes
  - One flag
    - Flag is not able to be a typo, since it is usually only one letter

Can We Generate Typos?



```
def swap_letters(word):
    typo_arr = []
    word_as_arr = string_to_list(word)
    for a in range(len(word_as_arr)):
        temp1 = word_as_arr[a]
        mistake_letter1 = random.choice(get_nearby_keys(temp1))
        word_as_arr[a] = mistake_letter1
        typo_arr.append(list_to_string(word_as_arr))
        try:
            b = random.choice([*range(a, len(word_as_arr))])
            temp2 = word_as_arr[b]
            mistake_letter2 = random.choice(get_nearby_keys(temp2))
            word_as_arr[b] = mistake_letter2
            typo_arr.append(list_to_string(word_as_arr))
            word_as_arr[b] = temp2
        except:
            #do nothing
            continue
        word_as_arr[a] = temp1
    return typo_arr
```

Can We Generate Typos?

input: ‘command’

['fommand', 'tommand', 'cpmmand', 'cpmmajd', 'cokmand', 'cokkand', 'comnand', 'comnahd', 'commwnd', 'commwne', 'commabd', 'commabe', 'commanr', 'commanf']

Can We Encode Typos?

Can We Encode Typos?

input: ‘cpmmajd’

[c, p, m, m, a, j, d]

Can We Encode Typos?

input: ‘cpmmajd’

[c, p, m, m, a, j, d]

[3, 16, 13, 13, 1, 10, 4]

Can We Encode Typos?

[3, 16, 13, 13, 1, 10, 4]

[3, 16, 13, 13, 1, 10, 4, …, X]

Can We Recognize Typos?

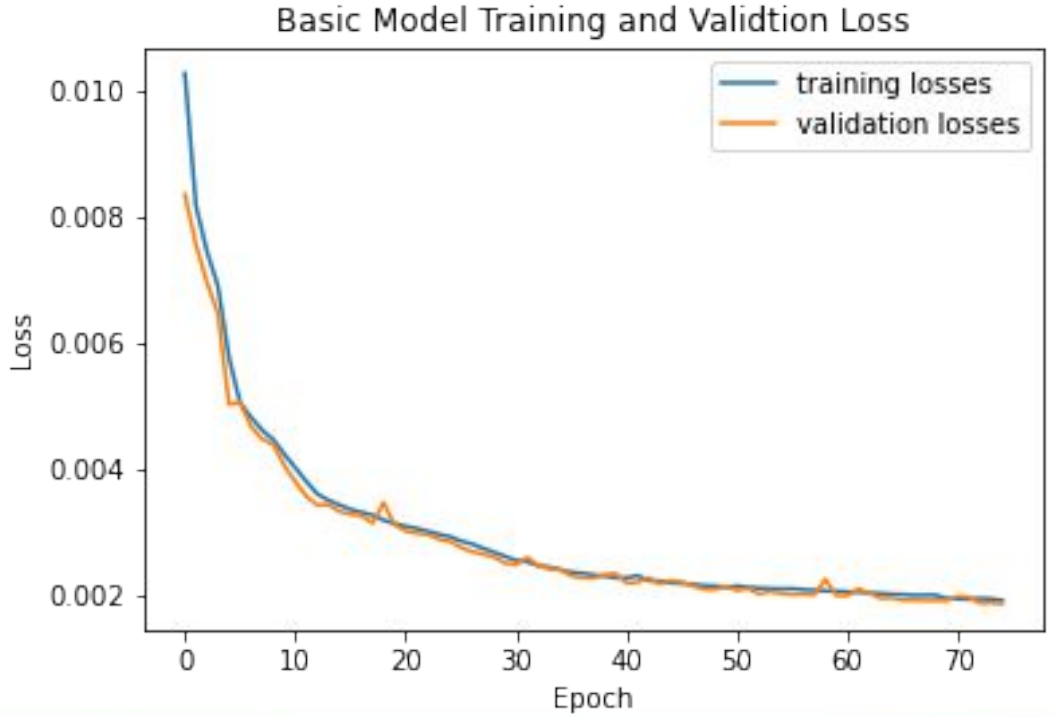
- Model structure: input size → larger hidden layer → less large hidden layer → output layer
  - Hidden layers ALWAYS larger than output layer
  - For my testing, I took a subset of 74 commands to classify

```python
# Create lists to hold training and val loss values
train_losses = []
val_losses = []

# Create lists to hold training and val accuracy values
train_accuracies = []
val_accuracies = []

# Train the network here
epochs = 75

for e in range(epochs):
    print("In epoch", e)
    running_loss = 0
    running_accuracies = []
    for inputs, labels in train_loader: # processing one batch at a time
        outputs = model(inputs) # predict labels
        loss = loss_criterion(outputs, labels) # calculate the loss

        # BACK PROPAGATION OF LOSS to generate updated weights
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        running_loss += loss.item()
        running_accuracies.append(get_accuracy(outputs, labels))

    running_val_loss = 0
    running_val_accuracies = []
    for val_inputs, val_labels in val_loader:
        val_outputs = model(val_inputs)
        running_val_accuracies.append(get_accuracy(val_outputs, val_labels))
        val_loss = loss_criterion(val_outputs, val_labels)
        running_val_loss += val_loss.item()

    print(f"\tTraining accuracy: {sum(running_accuracies)/len(running_accuracies)}")
    print(f"\tValidation accuracy: {sum(running_val_accuracies)/len(running_val_accuracies)}")
    print('')
    print(f"\tTraining loss: {running_loss/len(train_loader)}")
    print(f"\tValidation loss: {running_val_loss/len(val_loader)}")

    train_accuracies.append(sum(running_accuracies)/len(running_accuracies))
    val_accuracies.append(sum(running_val_accuracies)/len(running_val_accuracies))

    train_losses.append(running_loss/len(train_loader))
    val_losses.append(running_val_loss/len(val_loader))
```

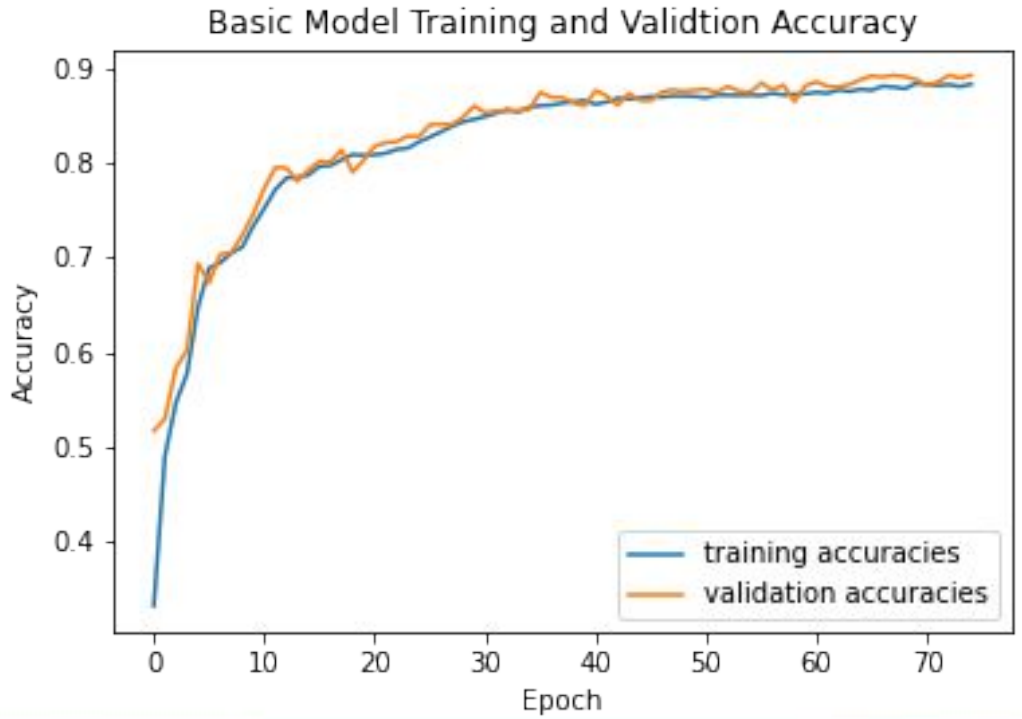
Can We Recognize Typos?

- Validation Loss: 0.008 → 0.002
- Validation Accuracy: 51.6% → 89.4%

Can We Recognize Typos?



Can We Recognize Typos?



THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC