

Copyright Notice

These slides are distributed under the Creative Commons License.

[DeepLearning.AI](#) makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite [DeepLearning.AI](#) as the source of the slides.

For the rest of the details of the license, see <https://creativecommons.org/licenses/by-sa/2.0/legalcode>



deeplearning.ai

Overview

Learning objectives

Learning objectives

- What is autocorrect?

Learning objectives

- What is autocorrect?
- Building the model

deah → dear ✓
yeah
dear
dean
... etc

Learning objectives

- What is autocorrect?
- Building the model
- Minimum edit distance

deah → dear ✓

yeah

dear

dean

... etc

| | # | s | t | a | y |
|---|---|---|---|---|---|
| # | 0 | 1 | 2 | 3 | 4 |
| p | 1 | 2 | 3 | 4 | 5 |
| l | 2 | 3 | 4 | 5 | 6 |
| a | 3 | 4 | 5 | 4 | 5 |
| y | 4 | 5 | 6 | 5 | 4 |

Learning objectives

- What is autocorrect?
- Building the model
- Minimum edit distance
- Minimum edit distance algorithm

deah → dear ✓

yeah

dear

dean

... etc

| | # | s | t | a | y |
|---|---|---|---|---|---|
| # | 0 | 1 | 2 | 3 | 4 |
| p | 1 | 2 | 3 | 4 | 5 |
| l | 2 | 3 | 4 | 5 | 6 |
| a | 3 | 4 | 5 | 4 | 5 |
| y | 4 | 5 | 6 | 5 | 4 |



deeplearning.ai

Autoccorrect

What is autocorrect?



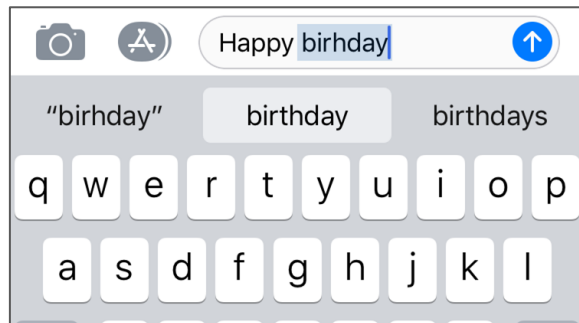
What is autocorrect?

- Phones



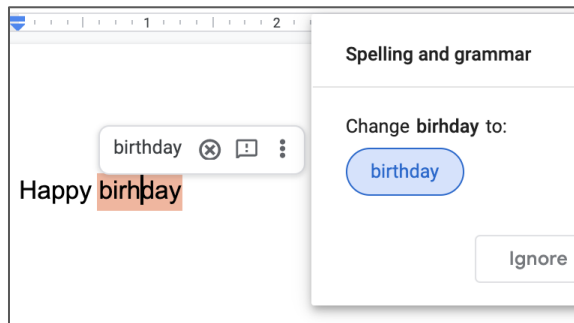
What is autocorrect?

- Phones
- Tablets



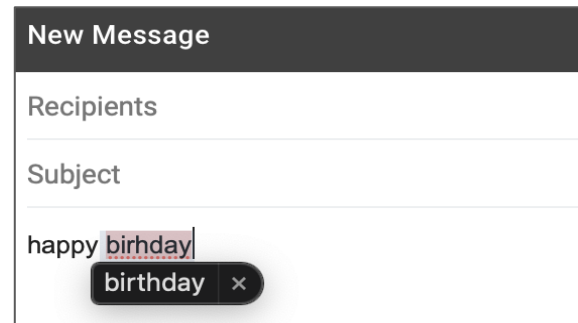
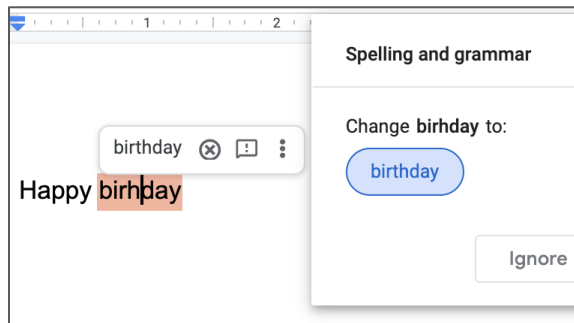
What is autocorrect?

- Phones
- Tablets
- Computers



What is autocorrect?

- Phones
- Tablets
- Computers



What is autocorrect?

- Example:

Happy birthday deah friend!



What is autocorrect ?

- Example:

Happy birthday dear friend!



What is autocorrect?

- Example:

Happy birthday deer friend!  ??

How it works

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

How it works

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

deah

How it works

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

deah
_eah
d_ar
de_r
... etc

How it works

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

deah
yeah
dear
dean
... *etc*

How it works

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

deah
yeah
| dear |
dean
... etc

How it works

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

deah → dear ✓
yeah
|dear|
dean
... etc



deeplearning.ai

Building the model

Building the model

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

Building the model

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

Building the model

1. Identify a misspelled word → if not in a dict

```
if word not in vocab:  
    misspelled = True
```

not meaning
or concept (just spelling)

deah ??



Building the model

1. Identify a misspelled word

```
if word not in vocab:  
    misspelled = True
```

deah

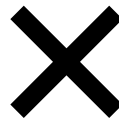


Building the model

1. Identify a misspelled word

```
if word not in vocab:  
    misspelled = True
```

deah



deer

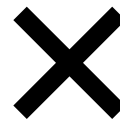


Building the model

1. Identify a misspelled word

```
if word not in vocab:  
    misspelled = True
```

deah



Happy birthday deer !



Building the model

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

Building the model

2. Find strings n edit distance away

Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it

Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
- Insert (add a letter)

Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
- Insert (add a letter)
 'to': 'top', 'two' ...



Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
- Insert (add a letter)
 'to': 'top', 'two' ...
- Delete (remove a letter)

Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
- Insert (add a letter)
 'to': 'top', 'two' ...
- Delete (remove a letter)
 'hat': 'ha', 'at', 'ht'



Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
- Insert (add a letter)
 'to': 'top', 'two' ...
- Delete (remove a letter)
 'hat': 'ha', 'at', 'ht'
- Switch (swap 2 adjacent letters)

Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
- Insert (add a letter)
'to': 'top', 'two' ...
- Delete (remove a letter)
'hat': 'ha', 'at', 'ht'
- Switch (swap 2 adjacent letters) 'eta': 'eat',
'tea'



Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
 - Insert (add a letter) 'to': 'top', 'two' ...
 - Delete (remove a letter) 'hat': 'ha', 'at', 'ht'
 - Switch (swap 2 adjacent letters) 'eta': 'eat', 'tea' 'ate'
- ✕

Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
- Insert (add a letter)
‘to’: ‘top’, ‘two’ ...
- Delete (remove a letter)
‘hat’: ‘ha’, ‘at’, ‘ht’
- Switch (swap 2 adjacent letters) ‘eta’: ‘eat’,
‘tea’



Building the model

2. Find strings n edit distance away

- Edit: an operation performed on a string to change it
- Insert (add a letter) 'to':
'top', 'two' ...
- Delete (remove a letter) 'hat':
'ha', 'at', 'ht'
- Switch (swap 2 adjacent letters) 'eta': 'eat',
'tea'
- Replace (change 1 letter to another) 'jaw': 'jar',



Building the model

2. Find strings n edit distance away

- Given a string find all possible strings that are n edit distance away using
 - Input
 - Delete
 - Switch (adjacent)
 - Replace

deah
_eah
d_ar
de_r
... etc

Building the model

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

Building the model

3. Filter candidates

*again based
on dict*

deah
_eah
d_ar
de_r
... etc

Building the model

3. Filter candidates

| | | |
|-------------|---|-------------|
| <u>deah</u> | | <u>deah</u> |
| _eah | | yeah |
| d_ar | → | dear |
| de_r | | dean |
| ... etc | | ... etc |



deeplearning.ai

Building the model II

Building the model

1. Identify a misspelled word
2. Find strings n edit distance away
3. Filter candidates
4. Calculate word probabilities

Building the model

4. Calculate word probabilities

*Find the most likely
candidate.*

Building the model

4. Calculate word probabilities

Example: “I am happy because I am learning”

| Word | Count |
|----------|-------|
| I | 2 |
| am | 2 |
| happy | 1 |
| because | 1 |
| learning | 1 |

Total : 7

Building the model

4. Calculate word probabilities

Example: "I am happy because I am learning"

Corpus

| Word | Count |
|----------|-------|
| I | 2 |
| am | 2 |
| happy | 1 |
| because | 1 |
| learning | 1 |

Total : 7

Building the model

4. Calculate word probabilities

Example: “I am happy because I am learning”

| Word | Count |
|----------|-------|
| I | 2 |
| am | 2 |
| happy | 1 |
| because | 1 |
| learning | 1 |

Total : 7

Building the model

4. Calculate word probabilities

Example: “I am happy because I am learning”

| Word | Count |
|----------|-------|
| I | 2 |
| am | 2 |
| happy | 1 |
| because | 1 |
| learning | 1 |

Total : 7

Building the model

4. Calculate word probabilities

Example: “I am happy because I am learning”

$$P(w) = \frac{C(w)}{V}$$

$P(w)$ Probability of a word

$C(w)$ Number of times the word appears

V Total size of the corpus

| Word | Count |
|----------|-------|
| I | 2 |
| am | 2 |
| happy | 1 |
| because | 1 |
| learning | 1 |

Total : 7

Building the model

4. Calculate word probabilities

Example: "I am happy because I am learning"

$$P(w) = \frac{C(w)}{V}$$

$$P(\text{am}) = \frac{C(\text{am})}{V} = \frac{2}{7}$$

| Word | Count |
|----------|-------|
| I | 2 |
| am | 2 |
| happy | 1 |
| because | 1 |
| learning | 1 |

Total : 7

$P(w)$ Probability of a word

$C(w)$ Number of times the word appears

V Total size of the corpus

Building the model

4. Calculate word probabilities

deah
yeah
[dear]
dean
... etc

Building the model

4. Calculate word probabilities

deah → dear ✓
yeah
dear
dean
... etc

Summary

1. Identify a misspelled word
2. Find strings n edit distance away

Insert
Delete
Switch
Replace

1. Filter candidates
2. Calculate word probabilities

$$P(w) = \frac{C(w)}{V}$$

deah → dear ☒
yeah
dear
dean
... etc

Summary

1. Identify a misspelled word
2. Find strings n edit distance away

Insert
Delete
Switch
Replace

1. Filter candidates
2. Calculate word probabilities

$$P(w) = \frac{C(w)}{M}$$

deah → dear ☒
yeah
dear
dean
... etc

Summary

1. Identify a misspelled word
2. Find strings n edit distance away

Insert
Delete
Switch
Replace

1. Filter candidates
2. Calculate word probabilities

$$P(w) = \frac{C(w)}{M}$$

deah → dear ☒
yeah
dear
dean
... etc

Summary

1. Identify a misspelled word
2. Find strings n edit distance away

Insert
Delete
Switch
Replace

1. Filter candidates
2. Calculate word probabilities

$$P(w) = \frac{C(w)}{M}$$

deah → dear ☒
_eah
d_ar
de_r
... etc

Summary

1. Identify a misspelled word
2. Find strings n edit distance away

Insert
Delete
Switch
Replace

1. Filter candidates
2. Calculate word probabilities

$$P(w) = \frac{C(w)}{M}$$

deah → dear ☒
yeah
dear
dean
... etc

Summary

1. Identify a misspelled word
2. Find strings n edit distance away

Insert
Delete
Switch
Replace

1. Filter candidates
2. Calculate word probabilities

$$P(w) = \frac{C(w)}{M}$$

deah → dear ☒
yeah
dear
dean
... etc

Summary

1. Identify a misspelled word
2. Find strings n edit distance away

Insert
Delete
Switch
Replace

1. Filter candidates
2. Calculate word probabilities

$$P(w) = \frac{C(w)}{M}$$

deah → dear ✓
yeah
dear
dean
... etc

Summary

1. Identify a misspelled word
2. Find strings n edit distance away

Insert
Delete
Switch
Replace

1. Filter candidates
2. Calculate word probabilities

$$P(w) = \frac{C(w)}{M}$$

deah → dear ✓
yeah
dear
dean
... etc



deeplearning.ai

Minimum edit distance

Minimum edit distance

- How to evaluate similarity between 2 strings?
- Minimum number of edits needed to transform 1 string into the other
- Spelling correction, document similarity, machine translation, DNA sequencing, and more

Minimum edit distance

- Edits:
- Insert (add a letter) 'to':
'top', 'two' ...
- Delete (remove a letter) 'hat': 'ha',
'at', 'ht'
- Replace (change 1 letter to another) 'jaw': 'jar', 'paw',
...

Minimum edit distance

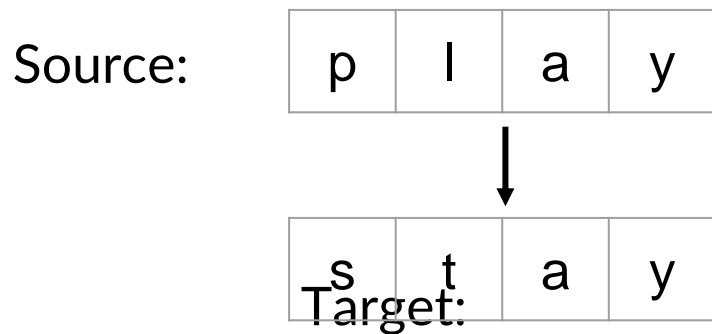
- Example:

Source:

| | | | |
|---|---|---|---|
| p | l | a | y |
|---|---|---|---|

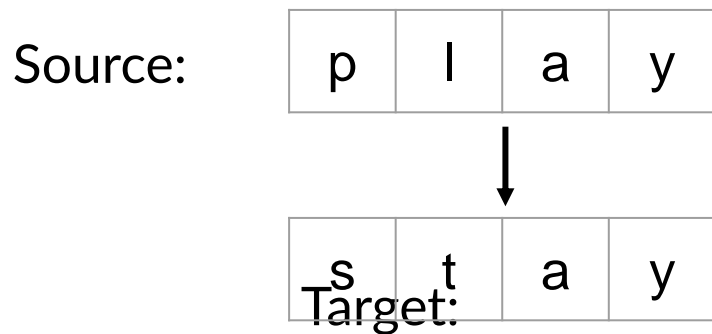
Minimum edit distance

- Example:



Minimum edit distance

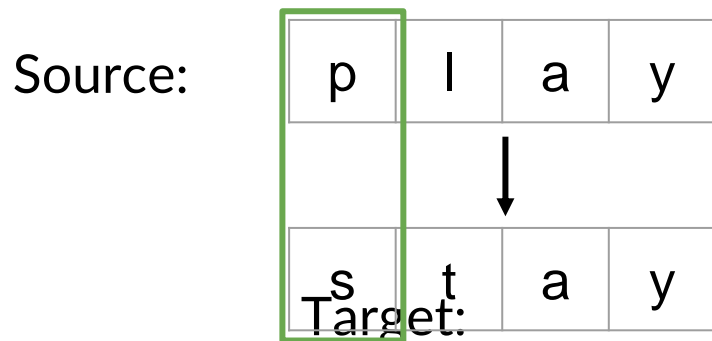
- Example:



What is the minimum number of edits to make this happen ?

Minimum edit distance

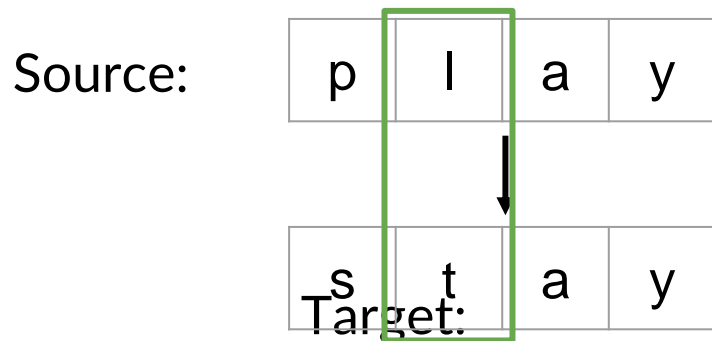
- Example:



$p \rightarrow s$: replace

Minimum edit distance

- Example:

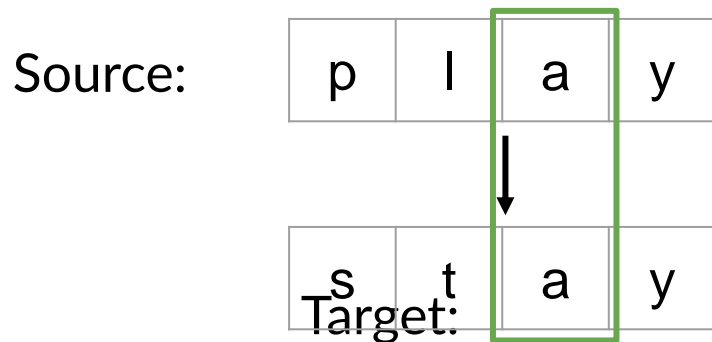


$p \rightarrow s$: replace

$l \rightarrow t$: replace

Minimum edit distance

- Example:

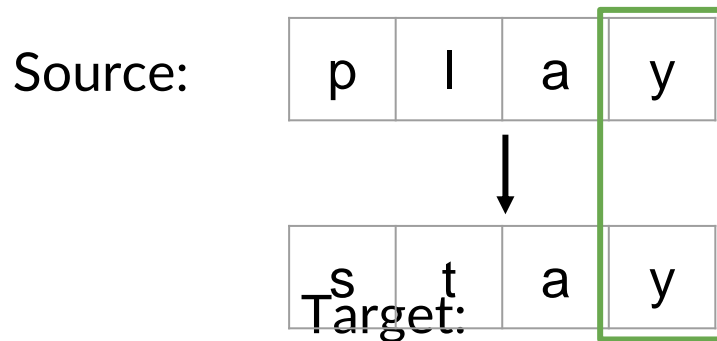


$p \rightarrow s$: replace

$l \rightarrow t$: replace

Minimum edit distance

- Example:

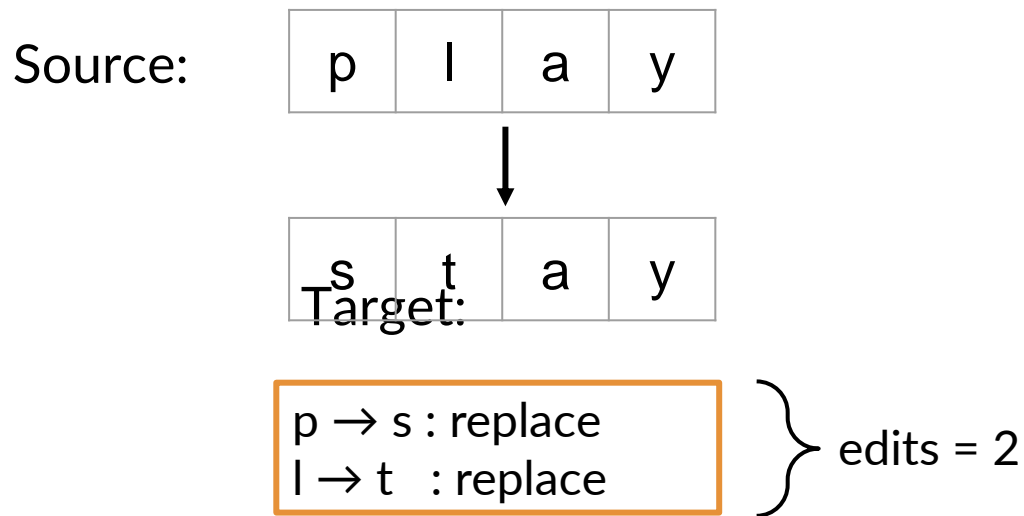


$p \rightarrow s$: replace

$l \rightarrow t$: replace

Minimum edit distance

- Example:



Minimum edit distance

- Example:

Source:

| | | | |
|---|---|---|---|
| p | l | a | y |
|---|---|---|---|



Target:

| | | | |
|---|---|---|---|
| s | t | a | y |
|---|---|---|---|

Edit cost:

Insert 1

Delete 1

Replace 2

| |
|-----------------|
| p → s : replace |
| l → t : replace |

} edits = 2

Minimum edit distance

- Example:

Source:

| | | | |
|---|---|---|---|
| p | l | a | y |
|---|---|---|---|



Target:

| | | | |
|---|---|---|---|
| s | t | a | y |
|---|---|---|---|

p → s : replace
l → t : replace

} edits = 2

Edit cost:

Insert 1

Delete 1

Replace 2

edit distance = 2 * 2 = 4

Minimum edit distance

- Example:

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | o | n | v | o | l | u | t | i | o | n | a | l | n | e | u | r | a | l | n | e | t | w | o | r | k |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Minimum edit distance

- Example:

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| c | o | n | v | o | l | u | t | i | o | n | a | l | n | e | u | r | a | l | n | e | t | w | o | r | k |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

CCAAGGGGTGACTCTAGTTTAATACTGAGATCAAATTATATGGGTGAT? !!



deeplearning.ai

Minimum edit distance algorithm

Minimum edit distance

Source: play → Target: stay

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

$D[]$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

$D[]$

$D[2,3] = pl \rightarrow sta$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

$D[]$

$D[2,3] = pl \rightarrow sta$

$D[2,3] = \text{source}[:2] \rightarrow \text{target}[:3]$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

$D[]$

$D[2,3] = pl \rightarrow sta$

$D[2,3] = \text{source}[:2] \rightarrow \text{target}[:3]$

$D[i,j] = \text{source}[:i] \rightarrow \text{target}[:j]$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play \rightarrow Target: stay

$D[]$

$D[i, j] = \text{source}[:i] \rightarrow \text{target}[:j]$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play \rightarrow Target: stay

$D[]$

$D[i, j] = \text{source}[:i] \rightarrow \text{target}[:j]$

$D[m, n] = \text{source} \rightarrow \text{target}$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play \rightarrow Target: stay

$D[]$

$D[i, j] = \text{source}[:i] \rightarrow \text{target}[:j]$

$D[m, n] = \text{source} \rightarrow \text{target}$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

The diagram illustrates the alignment between the source string 'play' and the target string 'stay' using a grid. The source string 'play' is represented by the first column of the grid (rows 0-4), and the target string 'stay' is represented by the first row of the grid (columns 0-4). The grid cells are colored blue for source characters and green for target characters. Three orange arrows indicate the alignment: a vertical arrow from row 0 to row 2 (representing 'p' to 'l'), a horizontal arrow from column 0 to column 2 (representing '#' to 't'), and a diagonal arrow from row 0, column 2 to row 2, column 0 (representing 'p' to 's').

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | | | | | |
| 1 | p | | | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

→

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| | # | | | | |
| 0 | # | | | | |
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

→

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | | | | |
| 0 | # | 0 | | | | |
| 1 | | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → #

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | | | | |
| 0 | # | 0 | | | | |
| 1 | p | | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → #
delete

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | | | | |
| 0 | # | 0 | | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

→ s

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

→ s
insert

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

$p \rightarrow s$

insert + delete: $p \rightarrow ps \rightarrow s$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

insert + delete: p → ps → s

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

insert + delete: p → ps → s:
2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play \rightarrow Target: stay

Cost: insert: 1, delete: 1, replace: 2

p \rightarrow s

insert + delete: p \rightarrow ps \rightarrow s:

2

delete + insert: p \rightarrow # \rightarrow s

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

insert + delete: p → ps → s:

2

delete + insert: p → # → s

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

insert + delete: p → ps → s:

2

delete + insert: p → # → s: 2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

insert + delete: p → ps → s:

2

delete + insert: p → # → s: 2

replace: p → s

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

insert + delete: p → ps → s:

2

delete + insert: p → # → s: 2

replace: p → s

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

insert + delete: p → ps → s:

2

delete + insert: p → # → s: 2

replace: p → s:

2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

insert + delete: p → ps → s:

2

delete + insert: p → # → s: 2

replace: p → s:

2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | | | |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | 2 | | | |
| 2 | | | | | | |
| 3 | | | | | | |
| 4 | | | | | | |



deeplearning.ai

Minimum edit distance algorithm II

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | 2 | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

play → #

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | 2 | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

play → #

$$D[i, j] = D[i-1, j] + del_cost$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | 2 | | | |
| 2 | l | | | | | |
| 3 | a | | | | | |
| 4 | y | | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

play → #

$$D[i, j] = D[i-1, j] + del_cost$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

play → #

$$D[i, j] = D[i-1, j] + del_cost$$

$$\begin{aligned} D[4, 0] &= \text{play} \rightarrow \# \\ &= \text{source}[:4] \rightarrow \text{target}[0] \end{aligned}$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

→ play

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | # | | s | t | a | y |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

→ play

$$D[i, j] = D[i, j-1] + ins_cost$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | | | |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

→ play

$$D[i, j] = D[i, j-1] + ins_cost$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i, j] = \begin{cases} D[i-1, j] + \text{del_cost} \\ \min \begin{cases} D[i, j-1] + \text{ins_cost} \\ D[i-1, j-1] + \begin{cases} \text{rep_cost}; & \text{if } \text{src}[i] \neq \text{tar}[j] \\ 0; & \text{if } \text{src}[i] = \text{tar}[j] \end{cases} \end{cases} \end{cases}$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i, j] = \begin{cases} D[i-1, j] + \text{del_cost} \\ \min \begin{cases} D[i, j-1] + \text{ins_cost} \\ D[i-1, j-1] + \begin{cases} \text{rep_cost}; & \text{if } \text{src}[i] \neq \text{tar}[j] \\ 0; & \text{if } \text{src}[i] = \text{tar}[j] \end{cases} \end{cases} \end{cases}$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i, j] = \min \begin{cases} D[i-1, j] + \text{del_cost} \\ D[i, j-1] + \text{ins_cost} \\ D[i-1, j-1] + \begin{cases} \text{rep_cost}; & \text{if } \text{src}[i] \neq \text{tar}[j] \\ 0; & \text{if } \text{src}[i] = \text{tar}[j] \end{cases} \end{cases}$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i, j] = \min \begin{cases} D[i-1, j] + \text{del_cost} \\ D[i, j-1] + \text{ins_cost} \\ D[i-1, j-1] + \begin{cases} \text{rep_cost}; & \text{if } \text{src}[i] \neq \text{tar}[j] \\ 0; & \text{if } \text{src}[i] = \text{tar}[j] \end{cases} \end{cases}$$

| | 0 | 1 | 2 | 3 | 4 | |
|---|---|---|---|---|---|---|
| | # | s | t | a | y | |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i, j] = \min \begin{cases} D[i-1, j] + \text{del_cost} \\ D[i, j-1] + \text{ins_cost} \\ D[i-1, j-1] + \begin{cases} \text{rep_cost; if } \text{src}[i] \neq \text{tar}[j] \\ 0; \text{ if } \text{src}[i] = \text{tar}[j] \end{cases} \end{cases}$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$D[i, j] =$

$$\min \begin{cases} D[i-1, j] + \text{del_cost} \\ D[i, j-1] + \text{ins_cost} \\ D[i-1, j-1] + \begin{cases} \text{rep_cost}; & \text{if } \text{src}[i] \neq \text{tar}[j] \\ 0; & \text{if } \text{src}[i] = \text{tar}[j] \end{cases} \end{cases}$$

dev purposes only

image of how previous slide should be

appearing for everyone !

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | # | s | t | a | y | |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Don't include text or images below this line. Delete this text and red line in the master template once you're finished with your slide creation

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

FORMULAS BUILDING ONLY

EQUATION USED IN NEXT SLIDES

$$D[i, j] = \min \begin{cases} D[i - 1, j] + del_cost \\ D[i, j - 1] + ins_cost \\ D[i - 1, j - 1] + \begin{cases} rep_cost; & \text{if } src[i] \neq tar[j] \\ 0; & \text{if } src[i] = tar[j] \end{cases} \end{cases}$$

$$D[i, j] = \min \begin{cases} D[i - 1, j] + del_cost \\ D[i, j - 1] + ins_cost \\ D[i - 1, j - 1] + \begin{cases} rep_cost; & \text{if } src[i] \neq tar[j] \\ 0; & \text{if } src[i] = tar[j] \end{cases} \end{cases}$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i-1, j] + 1 = 2$$

$$D[i, j-1] + 1 = 2$$

$$D[i-1, j-1] + 2 = 2$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i-1, j] + 1 = 2$$

$$D[i, j-1] + 1 = 2$$

$$D[i-1, j-1] + 2 = 2$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i-1, j] + 1 = 2$$

$$D[i, j-1] + 1 = 2$$

$$D[i-1, j-1] + 2 = 2$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$D[i-1, j] + 1 = 2$$

$$D[i, j-1] + 1 = 2$$

$$D[i-1, j-1] + 2 = 2$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

p → s

$$\begin{aligned} D[i-1, j] + 1 &= 2 \\ D[i, j-1] + 1 &= 2 \\ \left. \begin{array}{l} \\ \end{array} \right\} & \text{min} \\ = 2 \\ D[i-1, j-1] + 2 &= 2 \end{aligned}$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | | | |
| 2 | l | 2 | | | | |
| 3 | a | 3 | | | | |
| 4 | y | 4 | | | | |

Minimum edit distance

Source: to → Target: go

Cost: insert: 1, delete: 1, replace: 2

FOR QUIZ SETUP ONLY

... USED FOR IMAGES

ON QUIZ IN NEXT SLIDE

$$D[i, j] = \min \begin{cases} D[i-1, j] + \text{del_cost} \\ D[i, j-1] + \text{ins_cost} \\ D[i-1, j-1] + \begin{cases} \text{rep_cost}; & \text{if } \text{src}[i] \neq \text{tar}[j] \\ 0; & \text{if } \text{src}[i] = \text{tar}[j] \end{cases} \end{cases}$$

| | | 0 | 1 | 2 |
|---|---|---|---|---|
| | | # | g | o |
| 0 | # | 0 | 1 | 2 |
| 1 | t | 1 | 2 | 3 |
| 2 | o | 2 | 3 | |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | 3 | 4 | 5 |
| 2 | l | 2 | 3 | 4 | 5 | 6 |
| 3 | a | 3 | 4 | 5 | 4 | 5 |
| 4 | y | 4 | 5 | 6 | 5 | 4 |

Minimum edit distance

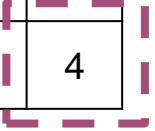
Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

play → stay

$$D[m, n] = 4$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | 3 | 4 | 5 |
| 2 | l | 2 | 3 | 4 | 5 | 6 |
| 3 | a | 3 | 4 | 5 | 4 | 5 |
| 4 | y | 4 | 5 | 6 | 5 | 4 |



Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

play → stay

$$D[m, n] = 4$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | 3 | 4 | 5 |
| 2 | l | 2 | 3 | 4 | 5 | 6 |
| 3 | a | 3 | 4 | 5 | 4 | 5 |
| 4 | y | 4 | 5 | 6 | 5 | 4 |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

play → stay

$$D[m, n] = 4$$

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | 3 | 4 | 5 |
| 2 | l | 2 | 3 | 4 | 5 | 6 |
| 3 | a | 3 | 4 | 5 | 4 | 5 |
| 4 | y | 4 | 5 | 6 | 5 | 4 |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | 3 | 4 | 5 |
| 2 | l | 2 | 3 | 4 | 5 | 6 |
| 3 | a | 3 | 4 | 5 | 4 | 5 |
| 4 | y | 4 | 5 | 6 | 5 | 4 |



deeplearning.ai

Minimum edit distance algorithm III

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

- Levenshtein distance

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | 3 | 4 | 5 |
| 2 | l | 2 | 3 | 4 | 5 | 6 |
| 3 | a | 3 | 4 | 5 | 4 | 5 |
| 4 | y | 4 | 5 | 6 | 5 | 4 |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

- Levenshtein distance
- Backtrace

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | 3 | 4 | 5 |
| 2 | l | 2 | 3 | 4 | 5 | 6 |
| 3 | a | 3 | 4 | 5 | 4 | 5 |
| 4 | y | 4 | 5 | 6 | 5 | 4 |

Minimum edit distance

Source: play → Target: stay

Cost: insert: 1, delete: 1, replace: 2

- Levenshtein distance
- Backtrace
- Dynamic programming

| | | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| | | # | s | t | a | y |
| 0 | # | 0 | 1 | 2 | 3 | 4 |
| 1 | p | 1 | 2 | 3 | 4 | 5 |
| 2 | l | 2 | 3 | 4 | 5 | 6 |
| 3 | a | 3 | 4 | 5 | 4 | 5 |
| 4 | y | 4 | 5 | 6 | 5 | 4 |



deeplearning.ai

Summary

Summary - learning objectives

- What is autocorrect ?
- Building the model
- Minimum edit distance
- Minimum edit distance algorithm

deah → dear ✓

yeah

dear

dean

... etc

| | # | s | t | a | y |
|---|---|---|---|---|---|
| # | 0 | 1 | 2 | 3 | 4 |
| p | 1 | 2 | 3 | 4 | 5 |
| l | 2 | 3 | 4 | 5 | 6 |
| a | 3 | 4 | 5 | 4 | 5 |
| y | 4 | 5 | 6 | 5 | 4 |