Hide menu

Lecture: Autocorrect and Minimum Edit Distance

- Video: Intro to Course 2 — 1 min
- Video: Week Introduction — 55 sec
- Video: Overview — 1 min
- Reading: Overview — 3 min
- Video: Autocorrect — 2 min
- Reading: Autocorrect — 4 min
- Video: Building the model — 4 min
- Reading: Building the model — 3 min
- Lab: Lecture notebook: Building the vocabulary — 1h
- Video: Building the model II — 3 min
- Reading: Building the model II — 4 min

Week 1 > Minimum edit distance III

Previous Next

Minimum edit distance III

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

To summarize, you have seen the levenshtein distance which specifies the cost per operation. If you need to reconstruct the path of how you got from one string to the other, you can use a backtrace. You should keep a simple pointer in each cell letting you know where you came from to get there. So you know the path taken across the table from the top left corner, to the bottom right corner. You can then reconstruct it.

This method for computation instead of brute force is a technique known as dynamic programming. You first solve the smallest subproblem first and then reusing that result you solve the next biggest subproblem, saving that result, reusing it again, and so on. This is exactly what you did by populating each cell from the top right to the bottom left. It's a well-known technique in computer science!