

# Autocorrect and Minimum Edit Distance

## How it works

- Identify misspelled word
- Find strings n distance away
- Filter candidates
- Calculate word probabilities

## Building the Model

1. Identify misspelled word – Is word in dictionary

If word not in vocab:

Misspelled = True

2. Find strings n edit distance away
    - a. Operation = insert, delete, replace, switch
  3. Filter candidates
    - a. Remove words not in vocabulary
  4. Calculate word probability
    - a. Find the word with the highest probability in corpus
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

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

Word	Count
I	2
am	2
happy	1
because	1
learning	1
Total: 7	

b.

## Minimum Edit Distance

- Min # of edits needed to transform String 1 to another
  - Spell correction, doc similarity, machine translation, DNA sequencing
  - Edits operation – insert, delete, replace (cost = 2 because it is delete then insert)
- 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				

- Measuring the edit distance by using the three edits: inserts, deletes, and replace with costs 1, 1, and 2 respectively is known as **Levenshtein** distance
- Dynamic Programming

Norvig's article - <https://norvig.com/spell-correct.html>

The goal of our spell check model is to compute the following probability:

$$P(c|w) = \frac{P(w|c) \times P(c)}{P(w)} \quad (\text{Eqn-1})$$

The equation above is [Bayes Rule](#).

- Equation 1 says that the probability of a word being correct  $P(c|w)$  is equal to the probability of having a certain word  $w$ , given that it is correct  $P(w|c)$ , multiplied by the probability of being correct in general  $P(C)$  divided by the probability of that word  $w$  appearing  $P(w)$  in general.