Autocorrect and Minimum Edit Distance

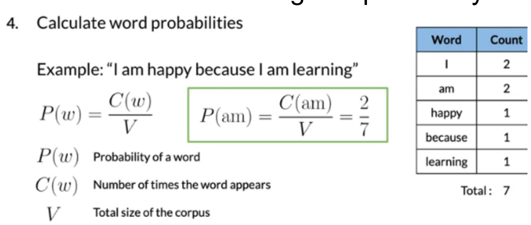
How it works

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

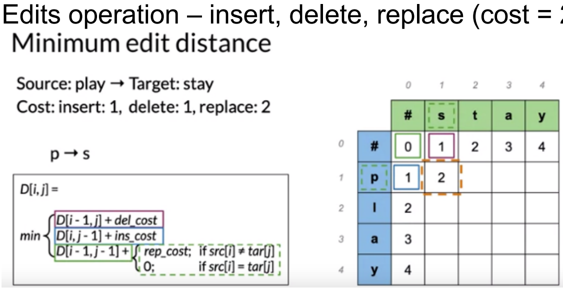
Building the Model

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
    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)
- 

- 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:

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

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.