# NLP \ THE GOOD PARTS

**HOW TO MAKE AN** 

#### <del>INTELLIGENT</del> SPELLING CORRECTOR

I'm Marco Herrero / @marhs\_

# PETER NORVIG

You can learn more in his web



Use the iPython notebook



Use your editor + python



Just drink beer and watch

```
> correct('madriz')
'madrid'
> correct('paella')
'paella'
> correct('qwweqqwe')
'qwweqqwe'
```

#### **IDEA**

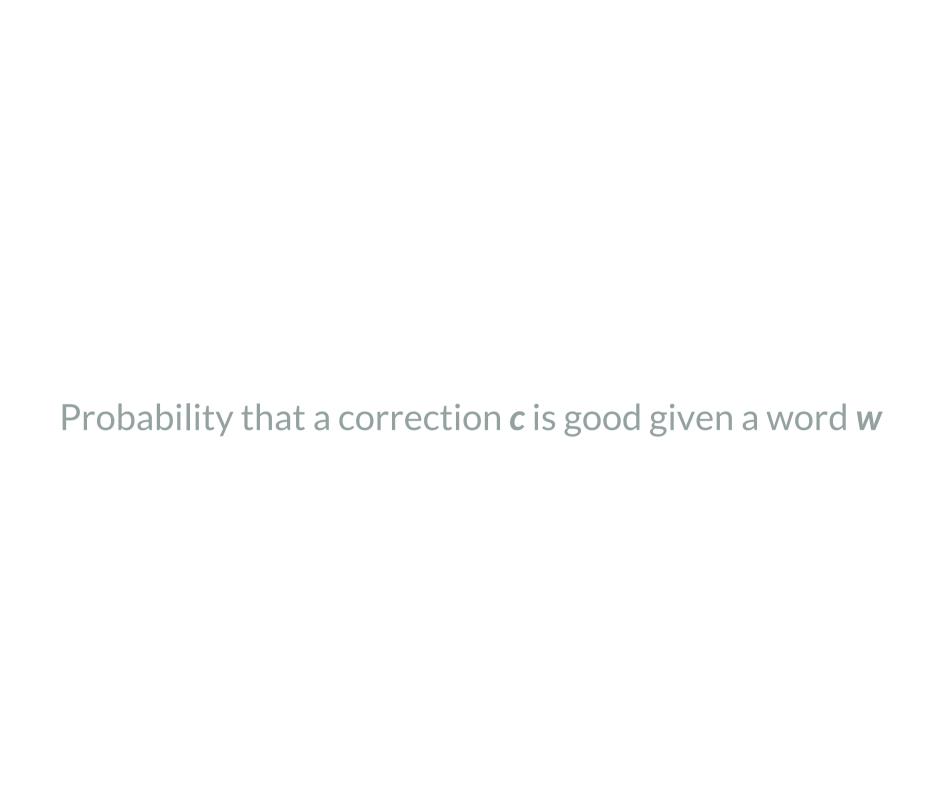
```
def correct(word):
    candidates = [] # ?
    return max(candidates, key="?") # ?
```

# A LITTLE PROBABILTY THEORY

(DON'T RUN)

#### **PROBLEM**

- Not sure about the correction
- We need to maximize probabilities



# P(C|W)

# P(RAIN UMBRELLA)

## $ARGMAX_{C} = P(C|W)$

```
return max(candidates, ...)
```

# BAYES' THEOREM P(C|W) = P(W|C) P(C) / P(W)

 $ARGMAX_{\mathbb{C}} = P(W|\mathbb{C}) P(\mathbb{C})$ 

#### - ARGMAXC

Control mechanism. Enumerate all c to choose the one with the best probability.

#### • P(C)

Language model. Prob. of the use of the correction.

### • P(W|C)

Error model. Prob. that w would be typed when the author meant c.

# LANGUAGE MODEL

Let's define the probability of each word with a dictionary and train the model with a text

Envia una push a todos los usuarios que hayan recibido una push.

Pushmaster, 2015.

#### NWORDS (LANGUAGE MODEL)

Word	Freq.	Value
push	2	3
recibido	1	2
XYXYXYYX	0	1

# ERROR MODEL

Probability of a mistake

#### EDIT DISTANCE

How many editions you need to do to go from

lati to later

2

lati to late

1

#### ERROR MODEL

P(known word) > P(edit=1) > P(edit=2) > P(invented word)

#### DEMO

#### **FUTURE WORK**

- Unicode
- Evaluation
- Language model of different size

# THE END

Thank you