

NLP \ THE GOOD PARTS

HOW TO MAKE AN INTELLIGENT SPELLING CORRECTOR

I'm [Marco Herrero](#) / [@marhs_](#)

PETER NORVIG

You can learn more in his web

YOU CAN



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="?") # ?
```


PROBLEM

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

Probability that a correction c is good given a word w

$$\text{ARGMAX}_C = P(C|W)$$

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


- **ARGMAX_c**

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

$P(\text{known word}) > P(\text{edit}=1) > P(\text{edit}=2) > P(\text{invented word})$

FUTURE WORK

- Unicode
- Evaluation
- Language model of different size

THE END

Thank you