Explicando un Corrector Ortográfico en 5 Minutos

Habla: Roberto Alsina
Código e Idea: Peter Norvig

Historia

- Hey Peter, el corrector ortográfico de Google está muy bueno, debe haber sido difícil!
- Pffffft

Teoría

Dada una palabra w que queremos corregir y un reemplazo c:

- La probabilidad de que hayas querido escribir c: P(c)

  Es más probable que quieras escribir "verde" y no "barde"

  Buscamos palabras **comunes**

- La probabilidad de que hayas escrito w cuando querías escribir c: P(w|c)

  Es más probable que si escribiste "berde" hayas querido poner "verde" y no "rojo"

  Buscamos palabras **parecidas**

Idealmente queremos palabras **comunes y parecidas**

Estimar P(c)

```
import re
from collections import Counter

def words(text): return re.findall(r'\w+', text.lower())

WORDS = Counter(words(open('big.txt').read()))

def P(word, N=sum(WORDS.values())):
    "Probability of `word`."
    return WORDS[word] / N
```

Estimar P(w|c)

Distancia de edicion!

Cuantas veces tengo que editar una palabra para convertirla en otra.

- hola -> bola = 1 (alteracion)
- hola -> ola = 1 (eliminacion)
- hola -> holas = 1 (insercion)
- hola -> ohla = 1 (transposicion)

Estimar P(w|c)

```
def edits1(word):
    "All edits that are one edit away from `word`."
    letters    = 'abcdefghijklmnopqrstuvwxyz'
    splits     = [(word[:i], word[i:]) for i in range(len(word) +
    deletes    = [L + R[1:] for L, R in splits if R]
    transposes = [L + R[1] + R[0] + R[2:] for L, R in splits if le
    replaces   = [L + c + R[1:] for L, R in splits if R for c in l
    inserts    = [L + c + R for L, R in splits for c in letters]
    return set(deletes + transposes + replaces + inserts)

def edits2(word):
    "All edits that are two edits away from `word`."
    return (e2 for e1 in edits1(word) for e2 in edits1(e1))
```

Estimar P(w|c)

Son demasiadas! ... Eliminamos las desconocidas.

```
def known(words):
    """The subset of `words` that appear in the
    dictionary of WORDS."""
    return set(w for w in words if w in WORDS)
```

ENTONCES?

```
def candidates(word):
    "Generate possible spelling corrections for word."
    return (known([word]) or known(edits1(word)) or
            known(edits2(word)) or [word])

def correction(word):
    "Most probable spelling correction for word."
    return max(candidates(word), key=P)
```