

Sistema de predicción de texto

Joaquín Jesús Pineda Gutiérrez

Luis Garrido Morillo

Problema

- ▶ El teclado de los SmartWatches agrupa las distintas letras del alfabeto de la siguiente manera, de forma que la escritura de palabras se realiza mediante predicción a partir de un análisis estadístico de las palabras más frecuentes.
- ▶ Esta predicción puede ser buena o mala dependiendo del conjunto de textos de referencia o corpus.



Modelos unigram y bigram

- ▶ La forma en la que se extrae la información probabilística dependiendo del modelo es la siguiente (considerando de ejemplo el texto “hola mundo”):
 - ▶ Caracteres unigram: {'a': 1, 'd': 1, 'h': 1, 'l': 1, 'm': 1, 'n': 1, 'o': 2, 'u': 1}
 - ▶ Caracteres bigram: {'a': {'l': 1}, 'd': {'n': 1}, 'l': {'o': 1}, 'n': {'u': 1}, 'o': {'d': 1, 'h': 1}, 'u': {'m': 1}}
 - ▶ Palabras unigram: {'hola': 1, 'mundo': 1}
 - ▶ Palabras bigram: {'mundo': {'hola': 1}}
- ▶ Toda esta información se guarda en distintos archivos dependiendo del modelo y si se trata de caracteres/palabras haciendo uso del módulo Pickle.

Modelos unigram y bigram

- ▶ A partir de los datos anteriores, se procesan en otros diccionarios usables para la predicción de texto a partir de la relación entre botones y caracteres, haciendo una traducción de los caracteres/palabras a combinación de botones y el número de apariciones. Por lo tanto, siguiendo el ejemplo anterior, los resultados serían:

- ▶ Caracteres unigram: {1: [('a', 1)], 2: [('d', 1)], 3: [('h', 1)], 4: [('l', 1)], 5: [('o', 2), ('m', 1), ('n', 1)], 7: [('u', 1)]}
- ▶ Caracteres bigram: {'a': {'4': [('l', 1.0)]}, 'd': {'5': [('n', 1.0)]}, 'l': {'5': [('o', 1.0)]}, 'n': {'7': [('u', 1.0)]}, 'o': {'2': [('d', 0.5)], '3': [('h', 0.5)]}, 'u': {'5': [('m', 1.0)]}}
- ▶ Palabras unigram: {'3541': [('hola', 1)], '57525': [('mundo', 1)]}
- ▶ Palabras bigram: {'mundo': {'3541': [('hola', 1.0)]}}

Relación entre
botones y
caracteres

1:	a,	á,	b,	c
2:	d,	e,	é,	f
3:	g,	h,	i,	í
4:	j,	k,	l	
5:	m,	n,	ñ,	o, ó
6:	p,	q,	r,	s
7:	t,	u,	ú,	v
8:	w,	x,	y,	z

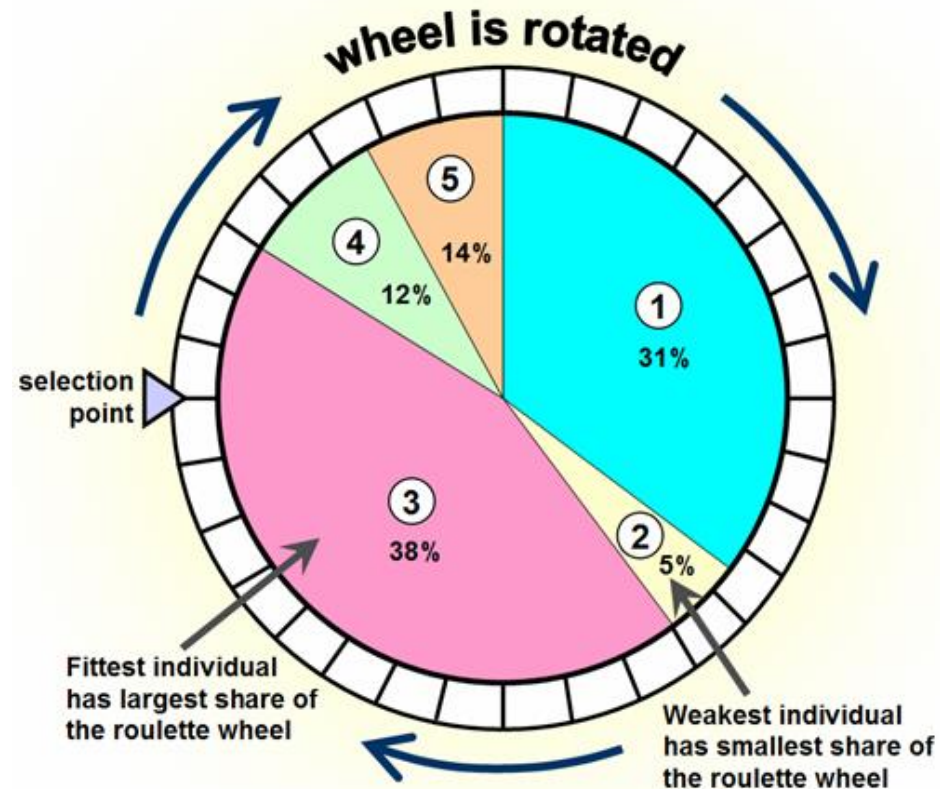
Extracción de datos

- ▶ En ambos modelos se realizan tratamientos para considerar mayúsculas, números o si son caracteres permitidos (los disponibles en el teclado). Sin embargo, en el modelo bigram se consideran más casos:
 - ▶ En la relación entre caracteres/palabras es importante la semántica, ya que dos palabras separadas por, por ejemplo, un punto no tienen relación directa.
 - ▶ Considerando esto, en caso de encontrar un signo de puntuación de los siguientes o un número, no se considera una relación entre las palabras separadas.
 - ▶ Signos de puntuación considerados: (,), ¡, ¿, ?, :, ;, @, #, “,”, ., ” “ (espacio, este se usa únicamente en bigram de caracteres), ..., _, -, [,], \

Predicción de texto

- ▶ En la predicción, se usan dos métodos distintos de selección de caracteres/palabras del corpus:

1. El que mayor número de apariciones tenga, obviando el resto de casos.
2. Método de la ruleta, dando ventaja a los casos más favorables sin ignorar al resto de casos.



Predicción de texto

- ▶ Con los datos extraídos de algunos textos de ejemplo y un texto a predecir como parámetro de entrada, la salida sería la siguiente:

```
Please, write a phrase: En un lugar de la Mancha cuyo nombre no quiero acordarme vivía
Here is the result:
Using words bigram
Phrase: 25 75 47316 22 41 515131 1785 555162 55 673265 115621652 73731
Prediction: en un lugar de la Mancha cuyo nombre no quiero acordarme vivía
Accuracy: 100.0 %
Here is the random result:
Using words bigram
Phrase: 25 75 47316 22 41 515131 1785 555162 55 673265 115621652 73731
Prediction: en un lugar de la Mancha cuyo nombre no quiero acordarme vivía
Accuracy: 100.0 %
```

- ▶ Siendo el primer método que se usa el de mayor aparición, y el segundo (random) el de la ruleta.
- ▶ Además, se expresa el acercamiento de la predicción al texto de entrada.

Interfaz gráfica

- ▶ Haciendo uso de tkinter, se muestra una ventana en la que se pueden ejecutar los distintos métodos y se resaltan los resultados:

