# PASO 1: Construcción de **WORDS** (para español)

1. Modificar el código para construir WORDS a partir de varios ficheros
2. Entrenar con ficheros de diccionarios españoles

* Para estos ficheros se asume que la palabra es correcta, y por tanto no se comprueba su existencia en la RAE.
* Algunos diccionarios españoles para entrenar aquí:
  + <http://pocketcm.com/phorum/read.php?14,1597>
  + <http://giusseppe.net/blog/archivo/2015/10/29/diccionario-de-la-rae-en-modo-texto-plano/>
  + <https://raw.githubusercontent.com/javierarce/palabras/master/listado-general.txt>
  + ~~<ftp://ftp.gnu.org/gnu/aspell/dict/0index.html>~~
  + <https://www.cs.hmc.edu/~geoff/ispell-dictionaries.html>
  + <http://icon.shef.ac.uk/Moby/mlang.html>
* ~~Algunos diccionarios inglés aquí:~~
  + ~~<http://www.mieliestronk.com/corncob_caps.txt>~~
  + ~~<https://lists.wordreference.com/>~~

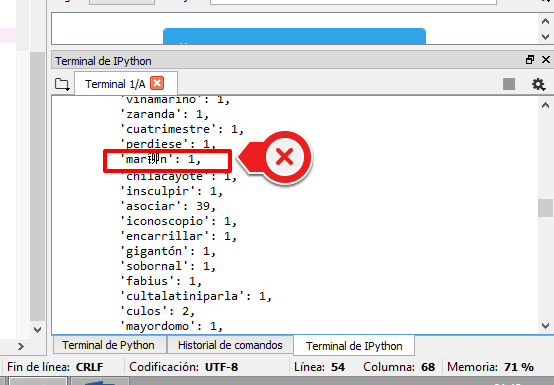
1. Añadir a WORDS las abreviaturas como palabras válidas

<http://www.rae.es/diccionario-panhispanico-de-dudas/apendices/abreviaturas>

1. Añadir a WORDS símbolos alfabetizables como palabras

<http://www.rae.es/diccionario-panhispanico-de-dudas/apendices/simbolos-alfabetizables>

1. Entrenar con varios corpus en español (corregidos los problemas de codificación)



Si da tiempo entrenar con mas corpus, me gustaría probar con otro que he encontrado que está construido de un libro y se llama spanish\_billion\_words)

* He tenido que cortar los ficheros corpus del parlamento porque se me ha quedado colgado el pc. Lo he cortado en trozos con los siguientes comandos UNIX

dos2unix europarl-v7.es-en.es

iconv -t UTF-8 europarl-v7.es-en.es > testUTF8.txt

mkdir trozos\_lineas

split -l 300 testUTF8.txt ./trozos\_lineas/trozo\_line

~~split -b 50000 europarl-v7\_short trozo\_50~~

* En este caso el fichero puede contener palabras incorrectas, y si añadimos a WORDS palabras inexistentes en el vocabulario, estaremos admitiendo como una palabra válida y le daremos una probabilidad > 0.
* Por tanto usaremos mecanismo adicional para determinar si se añade o no a WORDS. Integraremos con la API de la RAE (DRAE) para comprobar qué palabras son correctas y cuáles no. **(Abandono esta línea, la API ha dejado de funcionar, voy a investigar la API de wordreference)**
  + https://github.com/dialelo/rae-1/blob/master/README.rst
* <https://github.com/mgp25/RAE-API>
* <https://github.com/Telematica/RAE-Scrapper>
* <http://dle.rae.es/srv/fetch?w=ver%C3%ADa>
* Pruebo con otras APIs

<http://stackoverflow.com/questions/11039178/is-there-any-free-online-dictionary-api-json-xml-with-multiple-languages-to-ch>

<https://glosbe.com/gapi/translate?from=spa&dest=spa&format=json&phrase=hola>

En inglés

Este esta muy bien pero no está en español

<http://services.aonaware.com/DictService/DictService.asmx>

Tiene un servicio para ver si existe una palabra

<http://services.aonaware.com/DictService/DictService.asmx/Define?word=%hola>

1. 2 opciones según resultado del paso previo **(en esta versión añado siempre la palabra)**

* Opción 1: La palabra sí es correcta (existe en el diccionario)
  + Actualizamos WORDS
  + Añadimos la palabra si no existía ya o actualizamos el número de ocurrencias
* Opción 2: La palabra no es correcta (no está en el diccionario RAE)
  + No se añade la palabra a WORDS
  + Se podría ir añadiendo a una lista de palabras bajo estudio, para que el usuario o con algún otro mecanismo determinar si debe añadirse a WORDS o no

# PASO 2: Añadir nuevas transformaciones y reglas: método *edits3*

Basarse en los errores más comunes del español (<https://es.wikipedia.org/wiki/Wikipedia:Lista_de_errores_ortogr%C3%A1ficos_comunes/M%C3%A1quinas>)

* No usar m antes de p y b
* b y v
* h y sin h
* acento y sin acento
* g y j

# PASO 3: Palabras homófonas

Son palabras homófonas aquellas que se pronuncian igual pero se escriben de un modo diferente y tienen significados distintos.

En el español se dan las palabras homófonas por lo que hay que tenerlas en cuenta a la hora de implementar el corrector ya que se puede dar el caso de que tengamos un error ortográfico siendo la palabra correcta, por ejemplo, “Hay ahí niños” el corrector identificaría las palabras “hay” y “ahí” como correctas sin embargo estamos cometiendo dos faltas de ortografía, la frase correcta sería “Ahí hay niños”

En primer lugar se va a partir de un listado (PalabrasHomofonas.csv) con estos casos, este listado tiene tres columnas (h1, h2, h3) ya que hay casos en los que se presentan 3 palabras homófonas (hay, ahí, ay) aunque en la mayoría de los casos son sólo dos. Por otro lado también es necesario construir una lista que contenga en una única columna las palabras homófonas (ListaHomogonas.csv). Para la detección de estas faltas ortográficas se va a proceder como sigue:

1. Comprobar si la palabra es homófona a partir de ListaHomogonas.csv
2. En el caso de que lo sea hay que comprobar si el uso de esta es correcto o si es su homófona la correcta. Para ello se propone usar el clasificador de Naive Bayes que habrá que aplicarlo a cada conjunto de palabras homófonas para los que se usará PalabrasHomofonas.

Sean:

* H: el atributo nominal que toma como valores las palabras homófonas {h1,h2,h3}, normalmente será {h1,h2}
* C: variable discreta que indica el conjunto al que pertenece la palabra homófona (1,…,91)
* X1: atributo nominal que indica la palabra antecesora
* X2: atributo nominal que indica la palabra predecesora
* P1: Probabilidad de que ocurra H habiendo ocurrido X1 => P(H/X1)
* P2: Probabilidad de que ocurra H habiendo ocurrido X2 => P(H/X2)

A partir del conjunto de entrenamiento se generará esta información en un dataframe como podemos observar en la siguiente tabla:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **H** | **C** | **X1** | **X2** | **P1** | **P2** |
| baca | 3 | la | del | P(baca/la) | P(baca/del) |
| baca | 3 | tiene | para | P(baca/la) | P(baca/del) |
| vaca | 3 | la | come | P(vaca/la) | P(vaca/come) |
| vaca | 3 | mi | le | P(vaca/mi) | P(vaca/le) |
| riza | 10 | se | el | P(riza/se) | P(riza/el) |
| risa | 10 | esa | es | P(risa/esa) | P(risa/es) |

El problema que resuelve el clasificador de Naive Bayes es:

h\* = arg max i=1,2,3 P(X1/hi) P(X2/hi) P(hi)

Se tratará a cada conjunto de pablaras homófonas de forma independiente, esto es que para calcular las probabilidades sólo se tendrán en cuenta las palabras {h1,h2,h3} del mismo conjunto C

Hay que contemplar la posibilidad de que se nos presente algún caso donde P(X1/hi) ó P(X2/hi) sean igual a 0 debido a que en el dataframe generado con el conjunto de entrenamiento no se haya dado el caso. Es por ello que se aplicará la técnica de descuento de Laplace, esto es que se asume que todos los casos se dan al menos una vez.

# PASO 4: Definir comportamiento si no se encuentran coincidencias

La palabra no existe y tampoco encuentro candidatas en la lista ¿qué debe hacer el corrector?

# PASO 5: Aprendizaje del corrector

* Aprender del usuario
* Posibilidad del usuario para añadir nuevos términos a WORDS. Por ejemplo para añadir términos específicos de su sector o ámbito:
  + Python o py: si el usuario es informático
  + Palabras usadas en regiones específicas: por ejemplo términos cordobeses como pizco, perol…
* Prevalece la corrección que haya realizado el usuario, siempre se añade a WORDS ¿Tiene sentido dar un peso distinto a las palabras que el usuario haya añadido?
* Posibilidad de eliminar una palabra del diccionario: blacklist o algo asi, para que si vuelvo a entrenar no la vuelva a añadir.
* Mantener vivo WORDS: actualización de WORDS con cada texto que introduzca el usuario para corregir

# Estado del arte

* **Ispell**: <https://www.cs.hmc.edu/~geoff/ispell.html>
* **Aspell**: <http://aspell.net/>
* **Language tool**: <http://www.languagetool.org/>
* **Microsoft Word**: <http://office.microsoft.com/es-es/word/?CTT=97>
* **Google**

**http://www.datsi.fi.upm.es/~coes/interactivo/palabra.cgi**

# Otras posibles mejoras

* Integrar con el detector de lenguaje del primer ejercicio
* Añadir nuevos idiomas
* Tienen más peso los cambios de teclas próximas a la palabra que quiero corregir

# Referencias

* <http://norvig.com/spell-correct.html>
* <http://liceu.uab.cat/~joaquim/language_technology/NLP/PLN_aplicaciones.html>