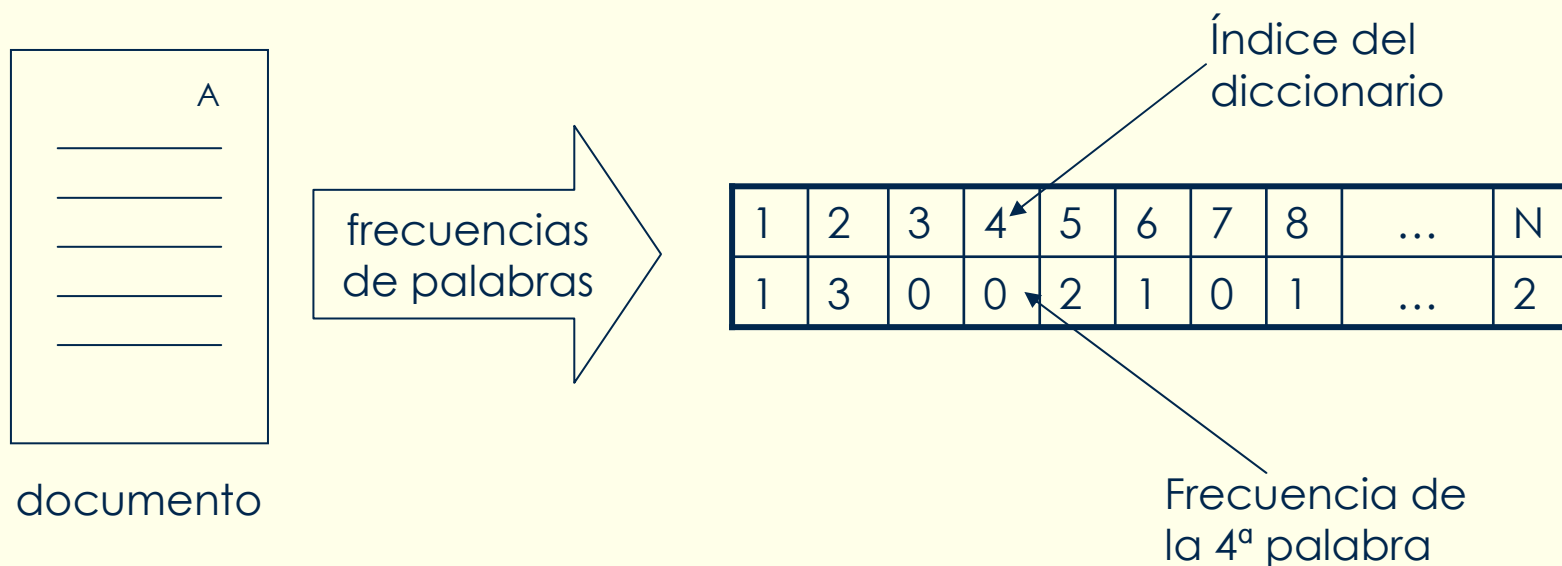




El modelo vectorial

# Un vector por documento



documento

## Selección de características:

- ¿Qué palabras tenemos que tener en cuenta?
- ¿Se pueden utilizar otros elementos?

## Cálculo de pesos:

- ¿Cómo calculamos la importancia de las características?

# Selección de características

- Escoger las palabras más frecuentes
- Elegir otro indicador para determinar qué palabras deben aparecer en el vector
- En lugar de utilizar como características palabras se pueden utilizar pares de palabras (p.e. *banco central*)
- Se pueden utilizar otras características más complejas como expresiones, sinónimos o cualquier elemento del texto que aporte información.

# Pesado por frecuencia relativa

Podemos introducir en la fórmula el número de palabras de cada documento, con lo que se reflejaría la importancia de la frecuencia relativa al tamaño del documento:

$$w_{jk} = 10 * \frac{1 + \log(\#(p_k, d_j))}{1 + \log(\#(d_j))}$$

$\#(p_k, d_j)$  es el número de veces que la palabra  $p_k$  aparece en el documento  $d_j$

$\#(d_j)$  es el número de palabras del documento  $d_j$

# Pesado *tfidf*

El pesado *tfidf* está pensado para contrarrestar el efecto negativo que tiene el hecho de que una palabra sea muy frecuente pero se encuentre en muchos documentos. La fórmula sería la siguiente:

$$\text{tfidf}(p_k, d_j) = \#(p_k, d_j) * \log \frac{\#D}{\#D(p_k)}$$

#D es el número total de documentos

#D( $p_k$ ) es el número de documentos que contienen la palabra  $p_k$

Se suele normalizar de la siguiente forma:

$$w_{kj} = \frac{\text{tfidf}(p_k, d_j)}{\sqrt{\sum_i (\text{tfidf}(p_i, d_j))^2}}$$



# Bigramas

Una forma de mejorar el rendimiento del clasificador es indexar los documentos por bigramas además de por palabras individuales: Algunos bigramas que pueden ser útiles en la clasificación de noticias son:

Real Madrid

Gobierno Español

déficit público

parrilla televisiva

Los bigramas a incluir en el modelo se seleccionarán de la misma forma que las palabras individuales. Por frecuencia o por tests de correlación.



## Ventanas (I)

La utilización de bigramas es una forma simple de incluir estructura sintáctica (sintagmas) en la indexación sin realizar un análisis sintáctico del texto. En algunas ocasiones se pueden "colar" otras palabras entre los extremos de un bigrama que hacen que la frecuencia de aparición se diluya entre varios ejemplos parecidos. Por ejemplo, para el bigrama:

Presidente Bush

que puede ser útil en la clasificación de noticias internacionales podemos encontrarnos con otros como:

Presidente George Bush

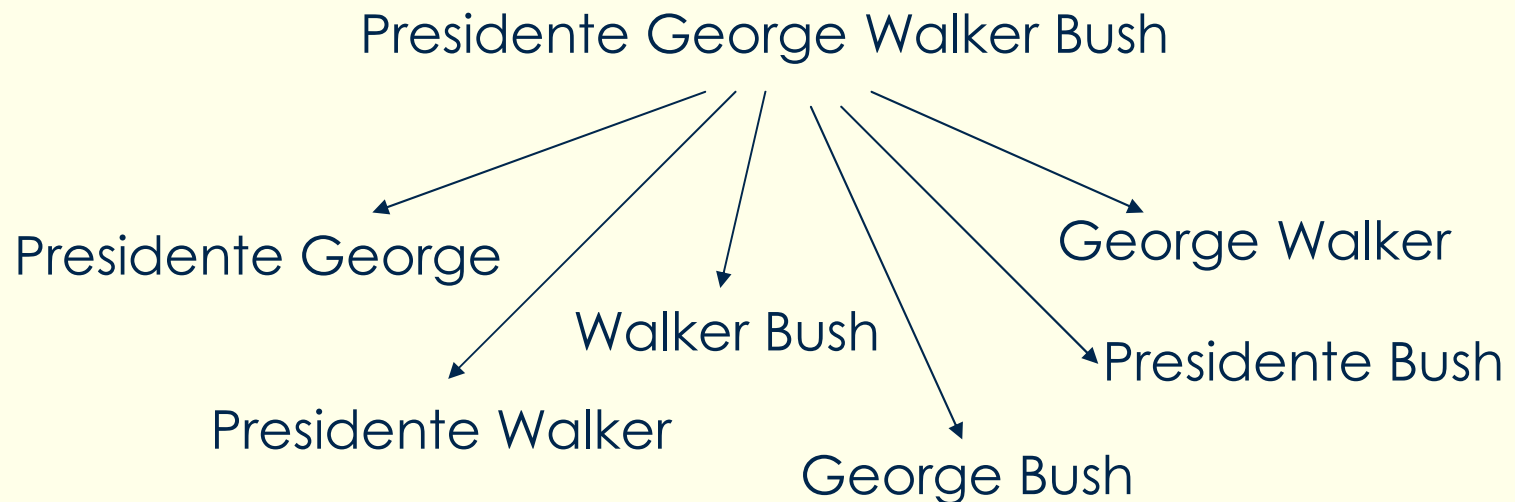
Presidente George W. Bush

Presidente G. W. Bush

Presidente George Walker Bush

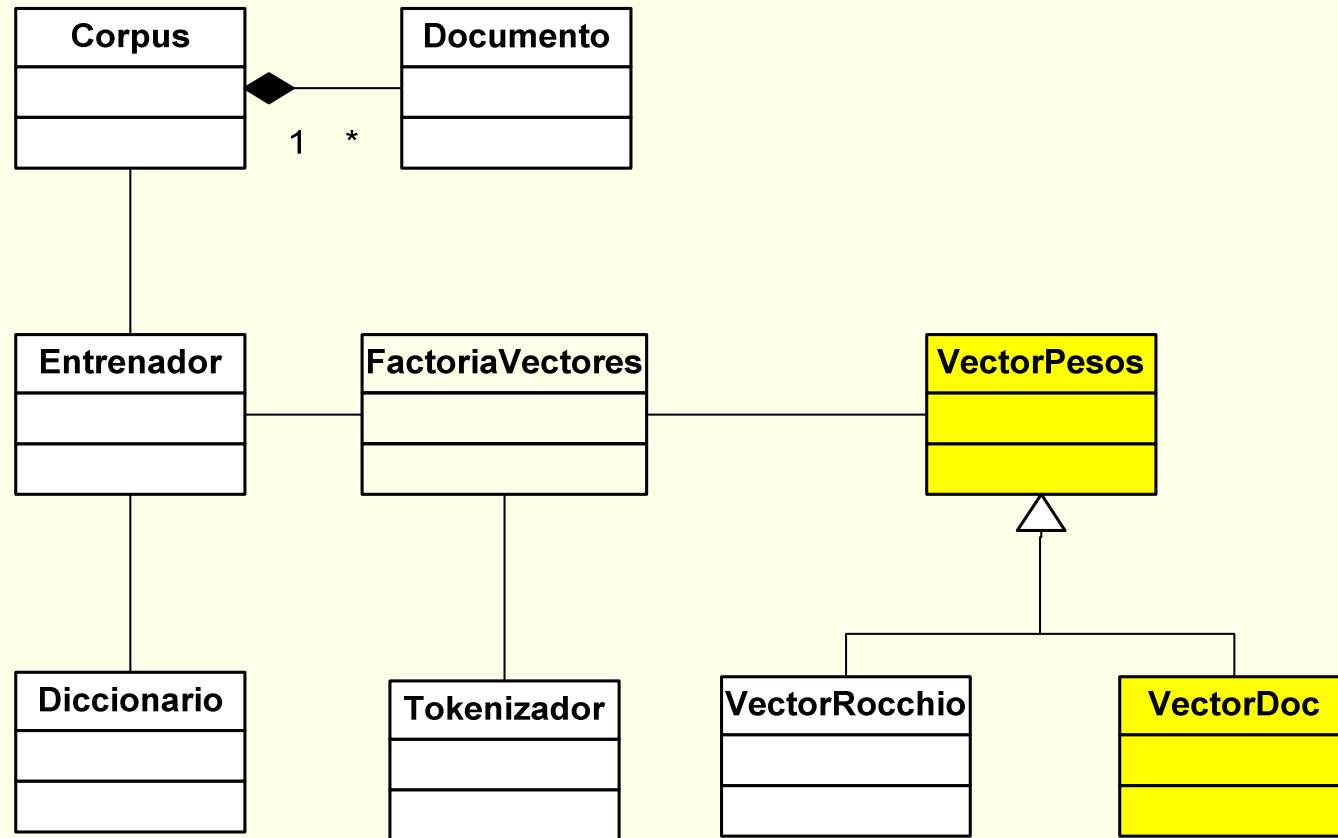
## Ventanas (II)

Para conseguir que todos los bigramas parecidos confluyan en uno sólo nos apoyaremos en el concepto de ventana. Por ejemplo utilizando una ventana de cuatro palabras podemos sacar seis bigramas distintos:



Ahora no se contarán los bigramas que aparezcan directamente sino aquellos que se puedan extraer de una ventana de cuatro palabras. De esta forma se conseguirá que los más frecuentes absorban las apariciones de aquellos que incluyen palabras extrañas.

# Clases a desarrollar



# La clase VectorPesos

```
/*
 * Número aproximado de líneas de código: 40
 */
public abstract class VectorPesos {
    public void guardaEnFichero(String fichero) {...}
    public int tamaño(){...}
    public double valor(int i){...}

    public double distanciaCoseno(VectorPesos v1){...}
    // No implementar aún
}
```

Más información en: [VectorPesos.html](#)

# La clase VectorDoc

```
/*  
 * Número aproximado de líneas de código: 60  
 */  
public class VectorDoc extends VectorPesos {  
    public VectorDoc(String texto, Diccionario dic,  
                    String tipo, Tokenizador tk) {...}  
    public VectorDoc(String fichero) {...}  
    public double[] vectorFrecuencias(String texto,  
                                     Tokenizador tk) {...}  
    public double[] vectorTfIdf(String texto,  
                                Tokenizador tk){...}  
    public double[] vectorTfIdfNorm(String texto,  
                                    Tokenizador tk){...}  
}
```

Más información en: [VectorDoc.html](#)