

Text Mining



Unidad 1 - Fundamentos de Text Mining

→ Contenidos

- ◆ Conceptos básicos
- ◆ Leyes del texto
- ◆ **Procesamiento de texto**
- ◆ **Problemas en Text Mining**



Unidad 1 - Fundamentos de Text Mining

Procesamiento de texto

Lo que hemos visto:

- Traspasado a minúscula
- Eliminar stopwords
- Eliminamos caracteres @, #, emojis, !, ?, ...
- Eliminar números
- Lematizar
- Stemming

Unidad 1 - Fundamentos de Text Mining

Stemming - Algoritmo de Porter

- Algoritmo de stemming muy utilizado en inglés
- Convenciones + 5 frases de reducción
- Las fases se aplican secuencialmente
- Cada fase consiste en un conjunto de reglas
 - Regla de ejemplo: eliminar la derivación *ement* si es largo del prefijo es mayor que 1
 - replacement → replac
- Convención de ejemplo: Si hay varias reglas que se pueden aplicar en un mismo caso, usar aquella que se aplica a un sufijo más largo.

Unidad 1 - Fundamentos de Text Mining

Stemming - Algoritmo de Porter

- Algunas reglas y ejemplos

Regla		Ejemplo	
SSES	→ SS	caresses	→ caress
IES	→ I	ponies	→ poni
SS	→ SS	caress	→ caress
S	→	cats	→ cat

Material basado en: <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>

Stemming - Algoritmo de Porter

Algorithm The Porter Stemming Algorithm

Input: An English word;

Output: The stem or original type of input word;

Algorithm:

Step 1: Distinguishing vowels and consonants by using the following rules:

- (1) Letters a, e, i, o, u are vowels;
- (2) The letter y has the following three cases:
 - (a) If y is the beginning of a word, it is judged as a consonant. e.g., y is a consonant in the word young;
 - (b) If the previous letter of y is a vowel, y is judged as a consonant. e.g., y is a consonant in the word boy;
 - (c) If the previous letter of y is a consonant, y is judged as a vowel. e.g., y is a vowel in the word fly.
- (3) All other letters except a, e, i, o, u, y are consonants.

Step 2: Processing words with -s, -ing and -ed suffixes by using the following rules:

- (1) Words ending with -s are treated as follows:
 - (a) If the word ends with -sses, then restore it to -ss. e.g., the word caresses should be restored to caress;
 - (b) If the word ends with -ies, then delete -es. e.g., cries becomes cri;
 - (c) If the word ends with -s and one of all letters before s is a vowel at least, consider the following two cases:
 - (i) if the vowel is adjacent the last s, the word will not change. e.g., the word gas is the original type and does not need to change;
 - (ii) Otherwise, delete the last letter s. e.g., gaps restore to gap.
- (2) If the word ends with -ing and the previous part of the word contains a vowel letter except for ing, delete ing. e.g., the word doing restore to do.

Step 3: Use the following rules to process words with other suffixes.

- (1) If the word ends with -y and the previous part of -y contains vowel letters, -y is changed to i. e.g., the word happy is rewritten as happi.
- (2) If the word ends with -ational and the previous section of -ational contains vowel letters, -ational is rewritten as -ate, for example, the word relational is rewritten as relate.

Step 4: Fine-tuning by the following rules:

For the words ending with e, if the number of consonants is greater than 1 except for the first letter and the last letter, the last letter e is removed, for example, relate is changed to relat.

Unidad 1 - Fundamentos de Text Mining

Texto de ejemplo: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Porter: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Lovins: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpre

Paice: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Unidad 1 - Fundamentos de Text Mining

Corrección ortográfica

- Dos usos principales
 - Corrección de documentos a ser indexados/consultados/...
 - Corrección
- Diferentes métodos para corrección ortográfica
 - Corrección de palabras aisladas
 - Chequear cada palabra con sígla misma (versión diccionario)
 - Desventaja: no identifica tipos, ej: un asteroide cayó del *celo*
 - Corrección sensible al contexto
 - Revisar una ventana de texto
 - Podrá detectar *celo*/*cielo*

Unidad 1 - Fundamentos de Text Mining

Corrección ortográfica

- Primero corrección ortográfica de palabras aisladas
- Premisa 1: Existe una lista de “palabras correctas” desde las cuales verificar.
- Premisa 2: Existe una forma de calcular distancia entre pares de palabras.
- Algoritmo simple de corrección: retornar las palabras “correctas” por distancia mínima.
- Ejemplo: informacón → información
- Se puede utilizar el vocabulario como la lista de palabras correctas.
- ¿Problema?

Unidad 1 - Fundamentos de Text Mining

Corrección ortográfica - Para una consulta

- Alternativas para vocabulario de referencias
 - Diccionario estándar (webster, etc.) – <https://www.merriam-webster.com/>
 - Diccionario específico (temático)
 - Vocabulario de la colección
 - Tesauro (permite representar conceptos)

Material basado en: <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>

Unidad 1 - Fundamentos de Text Mining

Corrección ortográfica - Distancia entre palabras

- Distancia Levenshtein
- Distancia Damerau-Levenshtein
- Distancia edición con pesos (weighted)
- Overlaps en k-gramas.

Unidad 1 - Fundamentos de Text Mining

Corrección ortográfica - Distancia entre palabras

- Distancia edición entre S1 y S2 es el número mínimo de operaciones básicas para convertir S1 en S2.
 - **Distancia Levenshtein:** similar a distancia edición, incluyendo operaciones admisibles como: insert, delete, replace.
 - **Damerau-Levenshtein:** incluye la transposición como cuarta operación admisible
cat - act : 1

Unidad 1 - Fundamentos de Text Mining

Corrección ortográfica - Distancia entre palabras

```
LEVENSHTEIN( $s_1, s_2$ )
1  for  $i \leftarrow 0$  to  $|s_1|$ 
2  do  $m[i, 0] = i$ 
3  for  $j \leftarrow 0$  to  $|s_2|$ 
4  do  $m[0, j] = j$ 
5  for  $i \leftarrow 1$  to  $|s_1|$ 
6  do for  $j \leftarrow 1$  to  $|s_2|$ 
7    do if  $s_1[i] = s_2[j]$ 
8      then  $m[i, j] = \min\{m[i-1, j] + 1, m[i, j-1] + 1, m[i-1, j-1]\}$ 
9      else  $m[i, j] = \min\{m[i-1, j] + 1, m[i, j-1] + 1, m[i-1, j-1] + 1\}$ 
10 return  $m[|s_1|, |s_2|]$ 
```

Operaciones

Insert (costo 1)

delete (costo 1)

replace (costo 1)

copy (costo 0)

Unidad 1 - Fundamentos de Text Mining

Ejemplo Matriz Levenshtein entre OSLO - SNOW

		s		n		o		w		
		0	1	1	2	2	3	3	4	4
o		1	1	2	2	3	2	4	4	5
		1	2	1	2	2	3	2	3	3
s		2	1	2	2	3	3	3	3	4
		2	3	1	2	2	3	3	4	3
l		3	3	2	2	3	4	4	4	4
		3	4	2	3	2	3	3	4	4
o		4	4	3	3	3	2	4	4	5
		4	5	3	4	3	4	2	3	3

cost	operation	input	output
1	delete	o	*
0	(copy)	s	s
1	replace	l	n
0	(copy)	o	o
1	insert	*	w

Material basado en: <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>

Unidad 1 - Fundamentos de Text Mining

Corrección ortográfica - Distancia entre palabras

Distancia edición con pesos (weighted)

- Similar a lo anterior, solo que ahora se entrega un peso a la operación, dependiendo del carácter involucrado.
- Trata de capturar errores de teclado, ej. m por n más que por q
- Además de reemplazar m por n tendrá una distancia edición menor que con q
- Se necesita una matriz de pesos como entrada.

Unidad 1 - Fundamentos de Text Mining

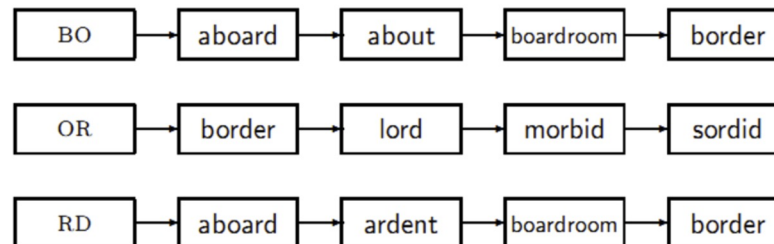
Corrección ortográfica - Distancia entre palabras

- Uso de la distancia edición
 - Dada una palabra/consulta, encontrar todas las posibles correcciones
 - Sugerir los términos que se encontraron al usuario (corrección on-line)
 - Realizar corrección automática puede tener riesgos (falsos positivos)

Unidad 1 - Fundamentos de Text Mining

Uso de k-gramas para ortografía

- Enumerar todos los k-gramas
- Utilizar un índice de k-gramas para recuperar las palabras correctas
- Considerar un umbral dado por el número de k-gramas que calzan
 - Ejemplo: términos que difieren a lo más en 3 k-gramas
 - Ejemplo bigramas - palabra mal escrita: bordroom
 - bo, or, rd, dr, ro, oo, om.



Material basado en: <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>

Unidad 1 - Fundamentos de Text Mining

Uso de k-gramas para ortografía

- Tema: si fijamos el número de k-gramas en que difieren dos palabras dejará de funcionar con palabras de largo distinto.
- Normalizar la noción de distancia con respecto al largo de las palabras comparadas.
- Ejemplo:
 - 1) NOVEMBER
 - Trigramas: NOV, OVE, VEM, **EMB**, **MBE**, **BER**
 - 2) DECEMBER
 - Trigramas DEC, ECE, CEM, **EMB**, **MBE**, **BER**

¿Es posible normalizar la medida de traslape?

Unidad 1 - Fundamentos de Text Mining

Coeficiente de Jaccard

- Medida de intersección entre dos conjuntos
- Sea A y B, se define como
$$\frac{|A \cap B|}{|A \cup B|}$$
- Tendremos un número entre 0 y 1
- Para corrección ortográfica calza, si por ejemplo se tiene un coeficiente > 0.8
- ¿Qué pasa si A y B tienen los mismo elementos?
- ¿Qué pasa si A y B son disjuntos?

Unidad 1 - Fundamentos de Text Mining

Soundex

- Idea: encontrar coincidencias fonéticas (no es muy utilizado)
- Ejemplo: chebyshev – tchebyscheff
- Algoritmo
 - Retener la primera letra del término
 - Cambiar todas las ocurrencias del término a 0 (cero): A, E, I, O, U, H, W, Y
 - Para las siguientes letras
 - B F P V a 1
 - C G J K Q S X Z a 2
 - D T a 3
 - L a 4
 - M N a 5
 - R a 6
 - Remover pares de dígitos contiguos si son similares
 - Eliminar todos los ceros del string resultante, recuperar los cuatro primeros caracteres.

Material basado en: <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>

Unidad 1 - Fundamentos de Text Mining

Soundex

- Ejemplo: HERMAN
 - Mantener la H
 - ERMAN → ORM0N
 - ORM0N → 06505
 - 06505 → 06505
 - 06505 → 655
 - return H655

¿HERMANN genera el mismo código?

Unidad 1 - Fundamentos de Text Mining

Problemas en Text Mining

- Clasificación de Texto (ej. reseña de productos)
- Análisis de sentimientos (ej. opiniones de clientes)
- Extracción de información (ej. resumen)
- Detección de temas (ej. agrupar temas similares)
- Análisis en redes sociales (ej. detección de rumores)
- Otros

Unidad 1 - Fundamentos de Text Mining

Problemas en Text Mining

¿Qué debemos considerar?

- Procesamiento de texto
 - Ruido en los datos, normalizar, stopword, etc.
- Palabras con diferentes significados
 - Sinónimos, polisemia, desambiguación semántica, etc
- Dimensionalidad
 - Muchos datos vs pocos datos
- Selección de Características
 - ¿Cuáles son las mejores características para representar mis datos?
- Lenguaje Natural
 - Coloquial vs formal
- Diferentes Lenguajes
- Etiquetado de Datos
- Privacidad y ética de los datos
- Otros factores

Material basado en: <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>

