Computació Intel·ligent I llenguatge Natural.

Práctica 1: Tagging basado en unigramas.

Presentación

Juan Soler Company
55.301 / 55.408
juan.soler@upf.edu

Contenido

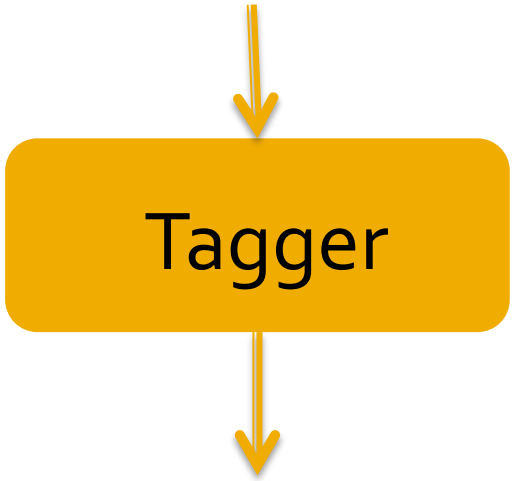
- Conceptos teóricos.
- Práctica 1.
- Temas logísticos.
- Problemas probables.

Conceptos Teóricos

- Un **Tagger** recibe una secuencia de palabras y les asigna una serie de etiquetas gramaticales (tags).

Conceptos Teóricos

El profesor es un crack.



DET N V DET ADJ

Conceptos Teóricos

- Dificultad del proceso de anotación/tagging: Ambigüedad!

- La categoría gramatical de una palabra depende del contexto.

  El profesor es muy crack/ADJ.
  El profesor fuma crack/N.

Conceptos Teóricos

- ¿Como automatizar este proceso?
  - Diccionario + reglas:
    - **If** word == "crack" **y** t(i-1) == V:

      Entonces t("crack") = N;

- Machine Learning.

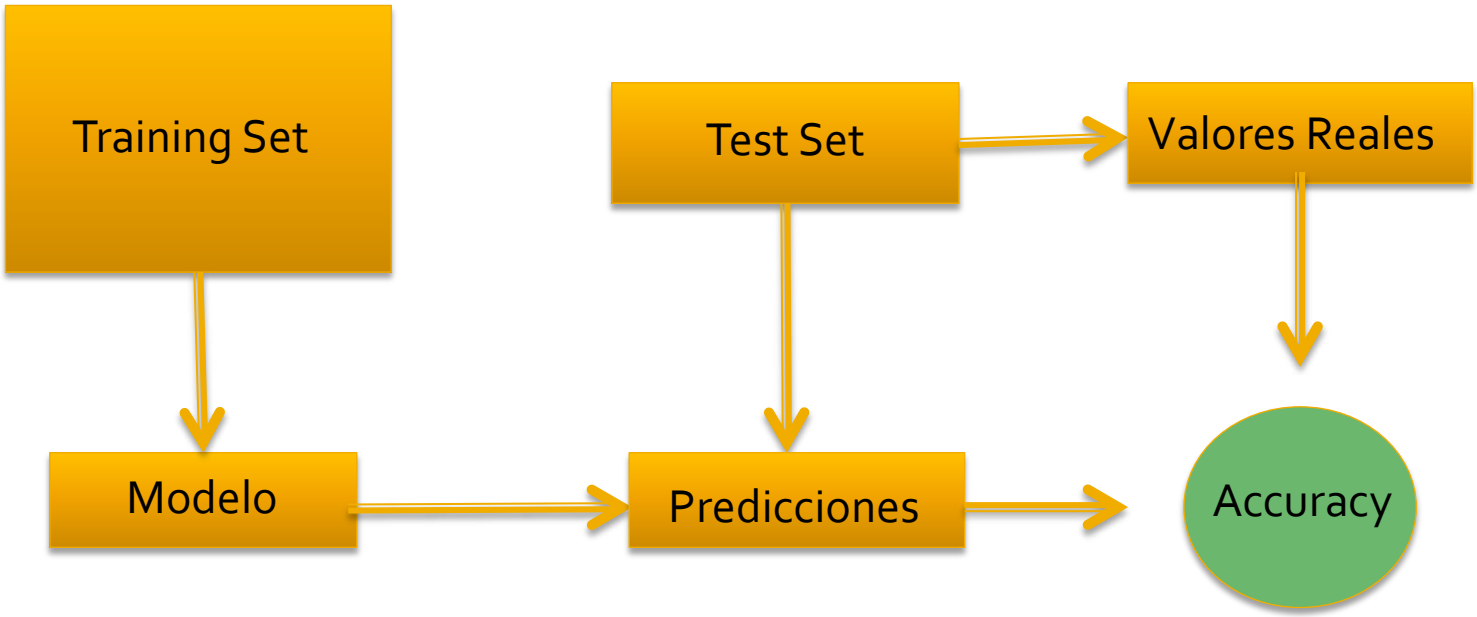
Conceptos Teóricos

Machine Learning?

Tagging como Supervised Machine Learning

- Dados una serie de datos anotados manualmente (Training set), extraer "conocimiento" para anotar nuevos textos.

Conceptos Teóricos



Anotación Estadística

- Definimos la probabilidad de que a una palabra W le corresponda un tag T como:

$$P(T|W)$$

Probabilidad Condicional:

$$P(A|B) = P(A \wedge B) / P(B)$$

Ej:

$P(t|w) = count(t,w) / count(w)$

Numerador: Veces que w aparece etiquetada como t.

Denominador: Veces que aparece la palabra w.

Implementaciones

- Etiquetado basado en unigramas.
  - Se asignarán las etiquetas más comunes de las palabras a etiquetar.
- Etiquetado basado en bigramas.
  - Se tendrá en cuenta el token anterior.

Tagging basado en unigramas

Dado un corpus de entrenamiento, al etiquetar una palabra, se asignará la etiqueta más común de esa palabra en el corpus de entrenamiento .

Ficheros

Corpus.txt -> Training set. Palabras anotadas manualmente.

Test_1/2.txt -> Test set. Palabras a anotar.

Gold_standard_1/2 -> Tags correctos para las palabras de test_1/2.txt

Pasos a seguir

1) Generación del modelo:

Dado el training set, escribir un programa que lo lea, y cuente para cada palabra y tag, su número de apariciones. Se deberá guardar el output de este programa en un fichero llamado "lexic.txt" con este formato:

Palabra	Tag	Apariciones
Cantar	V	440
Perro	N	330
Perro	ADJ	30

Pasos a seguir

2) Etiquetar utilizando el modelo.

Dados los ficheros de test, para cada palabra, asignarle el tag más probable según el modelo. Guardar el resultado en ficheros que tengan este formato para cada línea:

palabra predicción

Pasos a seguir

3) Evaluación de los resultados.

Comparar los outputs que se han generado con los ficheros gold_standard_1/2 y calcular la precisión en los dos casos:

accuracy = num pred correctas /num total

Temas Logísticos

Que entregar?

1- Código (40%): Comentado. Se valora claridad
2- Informe:

a) Explicar el proceso de etiquetado que se ha implementado y los resultados (30%)

b) En que dificultades se encuentra el programa con el fichero test_2? Como los solucionaríais? (30%)

Temas Logísticos

- Grupos de 2/3 personas.
- Lenguaje de programación libre (se recomienda PYTHON)
- Entrega: 12 Marzo 23:55.
- Informe en pdf.
- Entregar un zip con el código y el informe con nombre:
  - Nombre_apellido1_Nombre_apellido2.zip

Problemas probables

- Problemas con los acentos:
  - line = line.decode("latin_1").encode("UTF-8")
- El salto de línea en corpus.txt es un \n\r, no solo un \n. (cosas de Güindous).
- Dios mío, tarda mucho en recorrerme corpus.txt!!
  - Son más de 3 millones de líneas, dale a ejecutar y vete a tomar algo (procura que el código sea correcto)