Computación Inteligente i Lenguaje Natural

Práctica 2: Identificación de género usando bag-of-words classification

Contenido

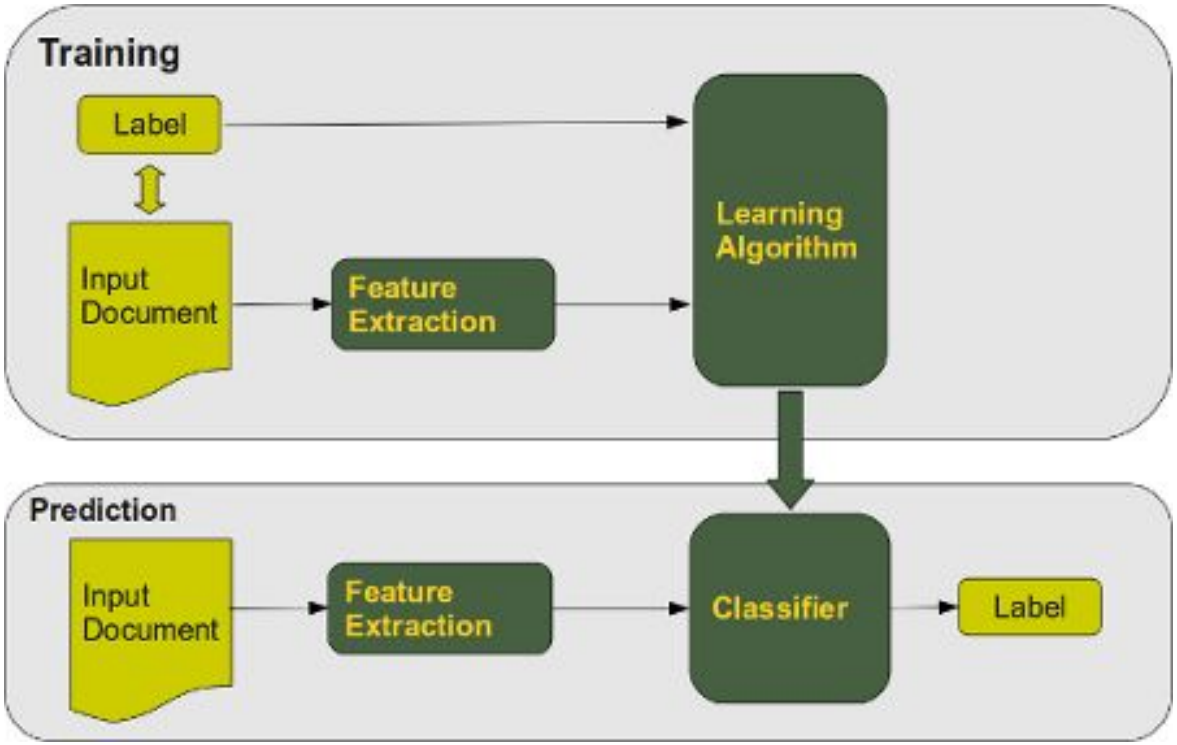
- Text Classification
- Text -> feature vectors
- Author profiling
- Bag-of-words
- Práctica 2

Text Classification

- Campo de investigación muy relevante.
- Dado un fichero de input, predecir la categoría a la que pertenece.
  - Ej: dado un texto, decidir si es de ficción o no.

- Feature Extraction y Machine Learning
- Gran cantidad de aplicaciones prácticas.

Text Classification



Text -> Feature Vectors

Dado un texto, se tiene que extraer una serie de características o features que caractericen su categoría o label y lo distingan de las otras categorías.

Es el proceso clave dentro de todo el desarrollo!

Text -> Feature Vectors

Ej:
"El Joan, ha resultat ser el millor professor de tots els temps, el seu carisma només és superat pel seu sex-appeal."
Label: Veritat

Possibles etiquetes: Veritat o Mentida
Feature a extreure: [num comes, num majúscules, num noms propis]

Vector de features: [2,2,1]

Text -> Feature Vectors

IMPORTANTE

Todos los vectores de features tienen que tener la misma longitud y cada feature debe estar en la misma dimensión.

Si nuestras features son las siguientes:
[comas, mayúsculas, nombres propios]

Cada vector deberá tener en la posición 0 el número de comas, en la 1 el número de mayúsculas y en la 2 el número de nombres propios.

Si una de estas características es 0, se pone un cero en la dimensión correspondiente:

NO SE PUEDE TENER VECTORES DE DIFERENTE MEDIDA

Author Profiling

Campo de investigación que intenta predecir rasgos demográficos de los autores de textos.

Principio básico: personas que comparten rasgos demográficos (género, edad, procedencia), también comparten rasgos lingüísticos que se pueden extraer e utilizar para predecir.

Author Profiling

CASO DE ESTUDIO:

Gender Identification, un subproblema dentro del author profiling.

Dado un texto, lo ha escrito un hombre o una mujer?

Bag of words

Estrategia simple pero efectiva.

Se usarán las frecuencias de las N palabras más frecuentes del corpus en cada uno de los textos, para clasificar según el género de los autores.

Bag of words

Ej:
Si N = 5 y las palabras más frecuentes son:
"I", "es", "yo", "tu", "ella"

El vector de cada instancia tendrá esta estructura:

[% "I", % "es", % "yo", % "tu", % "ella"]

Siendo cada dimensión, el porcentaje de las palabras del texto que corresponden a cada una de las palabras seleccionadas:

% "I" = # "I" en el texto / # palabras que tiene el texto

Práctica 2

Dataset/

Carpeta que contiene 1260 textos, en los que se indica en el nombre del fichero si los ha escrito un hombre, o una mujer:

1_male
2_female
…

Práctica 2 Pasos a seguir

1) Dado el corpus, extraer las N palabras más frecuentes.

2) Calcular los vectores de features para cada instancia:
Estos vectores tendrán N dimensiones con la frecuencia de cada palabra seleccionada en el texto en concreto.

Práctica 2 Pasos a seguir

3) Escribir los vectores de features en un formato entendible por algún toolkit de machine learning.

- WEKA
- scikit-learn

Calcular la precisión utilizando diferentes clasificadores.

Práctica 2 Pasos a seguir

3) WEKA

Weka necesita como input un fichero de tipo arff.
Tiene la siguiente pinta:

http://www.cs.waikato.ac.nz/ml/weka/arff.html

DEMO WEKA

Práctica 2 Pasos a seguir

3) scikit-learn

No tiene interfaz gráfica. Los clasificadores necesitan dos cosas:
X = [[vector inst 1], …, [vector inst N]]
Y = [label inst 1, …, label inst N]

MOSTRAR WEB SCIKIT LEARN

Práctica 2 Pasos a seguir

4) Variar los valores de N, el clasificador elegido y analizar cómo varía la precisión del sistema.

PASOS RECOMENDADOS

1) Python como lenguaje.
2) Sacar N palabras más frecuentes
3) Calcular vectores de features por cada instancia. Descargar Weka.
4) Pasar vectores a arff
5) Jugar con el weka (usar el explorer, probar con clasificadores como SMO, naive bayes, bagging, random forests...) haciendo 10-fold cross validation (es la opción por defecto).
6) Generar diversos arffs por cada valor de N y probar con diferentes clasificadores.

Entrega

Grupos 2-3
Lenguaje a elegir (mejor PYTHON)
Criterios evaluación:
- Extracción de N palabras frecuentes (10%)
- Cálculo features (35%)
- Generación arff /input scikit learn (15%)
- Resultados y análisis de los mismos variando N, clasificadores y mostrando la evolución (40%)
- PUNTOS EXTRA: Implementación de features extra que compitan o complementen el bag of words.

Entrega

Entrega 25 de Marzo 23:55

Por si acaso…

jUan.soler@upf.edu