Tipo : Guía de Laboratorio
Capítulo : NLP en Python
Duración : 60 minutos

I. OBJETIVO

- Ejecutar un caso de NLP centrado en encontrar la similitud entre un CV y perfil laboral usando NLTK y scikit-learn.

II. REQUISITOS

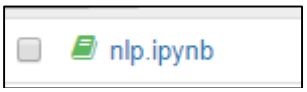
Los siguientes elementos de software son necesarios para la realización del laboratorio:

- Instalar Anaconda en Windows
- Navegador web

III. EJECUCIÓN DEL LABORATORIO

Ejercicio 3.2: NLTK y scikit-learn

- Crear un entorno virtual
    a. conda create --name nlplab python=3.5
    b. activate nlplab
    c. pip install
        i. jupyter
        ii. nltk
        iii. scikit-learn
        iv. pandas
        v. python-docx
- Activar jupyter en la línea de comandos con jupyter notebook
- Abrir nlp.ipynb en el browser
- Ejecutar el código y consultar



1. Leer achivo CSV jobs

```
In [4]: #Leer el archivo con descripciones de Bumeran
        import pandas as pd
        job_raw = pd.read_csv("bumall15oct.csv", encoding='cp1252')

In [5]: #visualizar la carga
        job_raw.head()
```

Out[5]:

	FECHA_SCRAP	CATEGORIA	FUNCION	EMPRESA	PUESTO	DESCRIPCION	URL	
0	15/10/2018 19:47	NaN	NaN	GSS CALL CENTER	Ejecutivo Ventas CALL CENTER C/EXP PRESENCIAL ...	TE INVITAMOS A SER PARTE DE NUESTRA FAMILIA GS...	http://www.bumeran.com.pe/empleos/ejecutivo-ve...	http://www.bumeran.com.pe/ep
1	15/10/2018 19:47	NaN	NaN	PANDERO S.A. EAFC	Jefe de Selección del Talento	Principales Funciones: - Responsable de los pr...	http://www.bumeran.com.pe/empleos/jefe-de-sele...	http://www.bumeran.com.pe/ep
2	15/10/2018 19:47	NaN	NaN	ANCRO	Administrador de Cuenta Comercial - Villa El S...	Administrador de Cuenta Comercial - Zona Sur E...	http://www.bumeran.com.pe/empleos/administrado...	http://www.bumeran.com.pe/ep

2. JOBS – data cleaning

```
#cargar la series con descripciones del puesto
desc = job_raw.DESCRIPCION
desc
```

```
0         TE INVITAMOS A SER PARTE DE NUESTRA FAMILIA GS...
1         Principales Funciones: - Responsable de los pr...
2         Administrador de Cuenta Comercial - Zona Sur E...
3         Funciones: - Generar la estrategia y el plan a...
4         La Gerencia de Mantenimiento Mayor de Latam Ar...
5         PACIFICO SEGUROS, empresa líder en el mercado ...
```

```
def desc_to_words(raw):
    #
    # 1. Solo letras
    letras = re.sub("[^a-zA-ZáóéíúñÑ]", " ", raw)
    # 2. convertir a minusculas
    words = letras.lower().split()
    #
    # 3. convertir a set ya que es más rapido
    stops = set(stopwords.words("spanish"))
    #
    # 4. Quitar stop words
    meaningful_words = [w for w in words if not w in stops]
    #
    # 5. Unir las palabras,
    # Retornar resultado.
    return( " ".join( meaningful_words ))
```

```
# Sacar el numero de registros
num_filas = desc.size
desc_limpio = []
for i in range(0, num_filas):
    desc_limpio.append(desc_to_words(desc[i]))
print ("COMPLETADO")
```

```
#convertir a pandas series
desc_limpio = pd.Series(desc_limpio)
print (desc_limpio)
```

3. JOBS – TF-IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(desc_limpio)
print (tfidf_matrix.shape)
```

```
(12659, 32282)
```

4. CV – carga

```
#cargar CV en word
import docx

def getText(filename):
    doc = docx.Document(filename)
    fullText = []
    for para in doc.paragraphs:
        fullText.append(para.text)
    return '\n'.join(fullText)

read_word = getText('CV_Abogado.docx')
```

5. CV – limpieza

```
# Probar con expresiones regulares
import re
#Solo quiero letras
letrascv = re.sub("[^a-zA-ZáóéíúñÑ]", " ", read_word)
```

```
# Pasar a minusculas
minusculascv = letrascv.lower()
palabrascv = minusculascv.split()
```

```
# Quitar los stopwords de la lista
import nltk
from nltk.corpus import stopwords
palabrascv = [w for w in palabrascv if not w in stopwords.words("spanish")]
```

```
#juntar todo
resultadocv = " ".join(palabrascv)
```

```
import pandas as pd
#convertir a pandas series
desc_limpio_cv = pd.Series(resultadocv)
print (desc_limpio_cv)
```

```
0    ejemplo cv alberto sala gonzalez c heroes alca...
```

6. CV – TF-IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer
#tfidf_vectorizer2 = TfidfVectorizer()
tfidf_matrix2 = tfidf_vectorizer.transform(desc_limpio_cv)
print (tfidf_matrix2.shape)
```

```
(1, 32282)
```

7. Distancia coseno

```
#Comparar el 1er documento al resto
from sklearn.metrics.pairwise import cosine_similarity
#res = cosine_similarity(tfidf_matrix[0:1], tfidf_matrix, True)
res = cosine_similarity(tfidf_matrix2, tfidf_matrix, True)
res = sorted(res[0], reverse=True)
res
```

```
#obtener el nombre del documento y crear un dataframe
res = cosine_similarity(tfidf_matrix2, tfidf_matrix, True)
res = res[0]
```

```
size = len(res)
#crear el dataframe
job_simil = pd.DataFrame(columns=('ID', 'Puesto', 'Similitud'))
#job_simil
i = int()
#llenar los datos
for i in range(0, size):
    job_simil.loc[i] = [i+1, job_raw['PUESTO'][i],res[i]]
#hacer un sort por similitud
sorted_job = job_simil.sort_values(['Similitud'], ascending=False)
```

```
sorted_job
```

	ID	Puesto	Similitud
3852	3853	Abogado - Asesor jurídico de control interno	0.146154
3483	3484	Secretaria Bilingüe	0.135361
7270	7271	DOCENTE TP–CURSO DERECHO DE LA PROPIEDAD INTEL...	0.128526

IV. EVALUACIÓN

1. ¿Cuáles son los pasos principales de NLP realizados en este caso?

    a. **Respuesta:**

- **Jobs**
  - Carga
  - Limpieza de datos
  - TF-IF
- **CV**
  - Carga
  - Limpieza de datos – preparación para TF-IDF
  - TF-IDF
- Se observa que el modelo se entrena, a partir de los Jobs (vocabulario principal).
- Distancia coseno para recomendar Jobs a partir del CV.