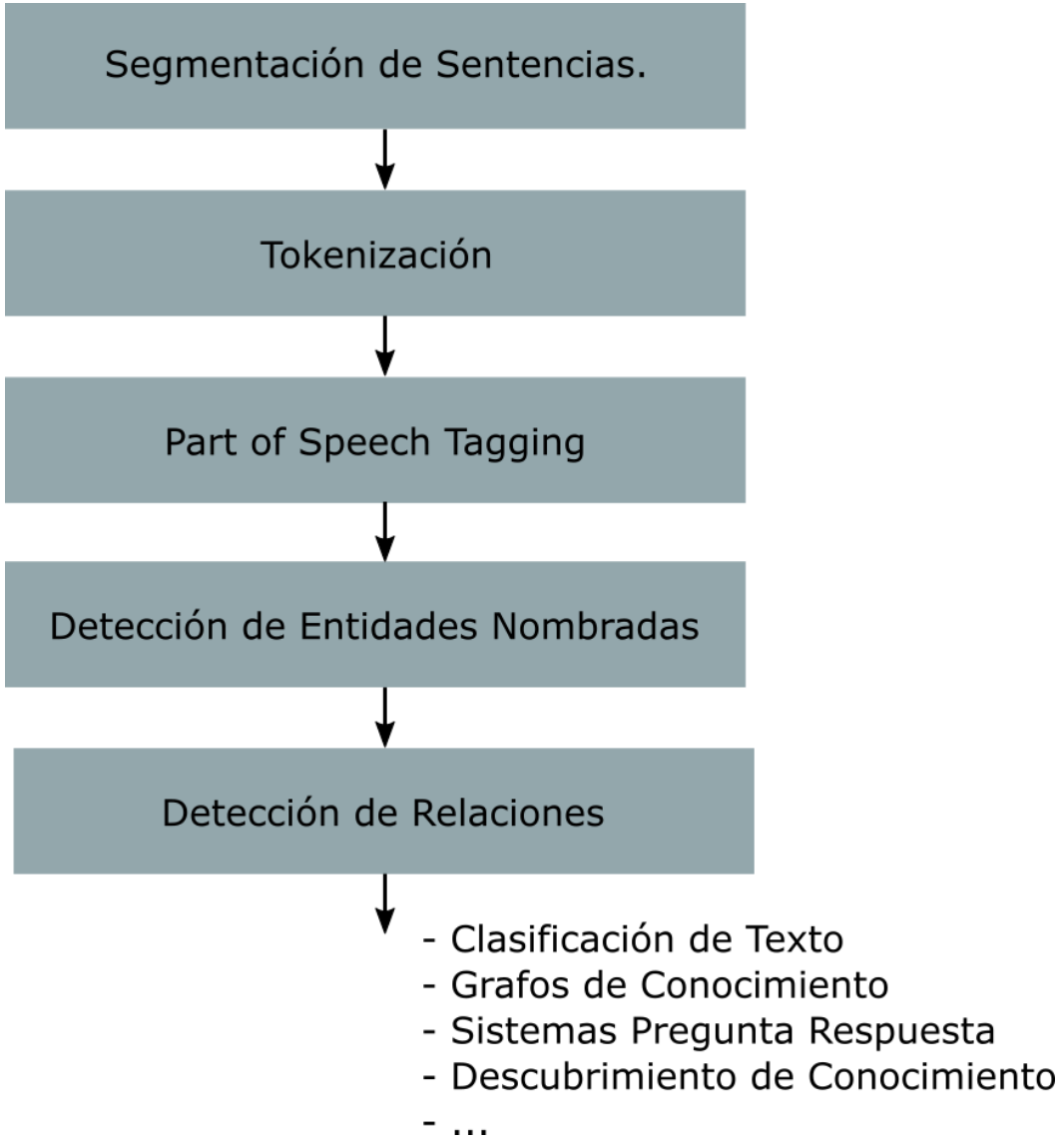
Herramientas de Extracción (Parte 3)

Dr. Cristian Enrique Muñoz Villalobos

Introducción

Existen varias estrategias para la extracción de información. Muchos de ellos basado en IA como también (proveniente del area linguística), basados en estructuras gramáticas y reglas sintácticas. De forma general presentamos el Pipeline del proceso de extracción de información:



Segmentación de Sentencias

- Se reduce a la tarea de dividir un texto en una composición de sentencias/oración (cadena mínima de caracteres que consiguen transmitir una idea).
- En varios lenguajes la puntuación es un aproximador razonable. Sin embargo, existe muchas situaciones (abreviaturas, códigos, "..." , ":", etc.) que dificultan la tarea de segmentación.
- La complejidad de esta tarea aumenta con la existencia de tablas formulas, formatos diferentes, etc.
- Existen bibliotecas que realizan la segmentación de sentencias. Muchos de ellos basados en modelos de IA y reglas.

Tokenización

- La tokenización segmenta una sentencia en palabras o unidades atómicas (**token**). Esta etapa es fundamental tanto en métodos tradicionales y avanzados de NLP.
- Realizada la tokenización, obtenemos una lista de tokens con los cuales creamos un vocabulario. Este vocabulario nos permite indexar cada token con la finalidad de utilizarlo en modelos de machine learning.

Se descarto que exista presecución política

Vocabulario

Se
descarto
que
exista
presecución
política

1
0

Tipos de Tokenización

La estrategia de tokenización puede ser establecida mediante un conjunto de reglas o es aprendido mediante un entrenamiento. De forma general podemos clasificarlas en 4 tipos:

- Tokenización Simple
- Tokenización basada en palabra
- Tokenización basada en caracteres
- Tokenización basada en sub-palabras

Tokenización Simple

Esta tokenización sigue a simple idea de definir como **token** el contenido que se encuentra entre espacios. Por ejemplo:

Dada la frase:

```
Se descarto, por ahora, que exista persecución política contra el.
```

Los tokens son: `Se` `descarto,` `por` `ahora,` `que` `exista` `persecición` `política` `contra` `el.`

Tokenización basada en palabras

Utilizada cuando se tiene la necesidad de conservar la interpretación sintáctica del texto.Esto facilita revision y lectura por parte del ser humano.

Dada la frase:

```
Se descarto, por ahora, que exista persecución política contra el.
```

Los tokens son: `Se` `descarto` `,` `por` `ahora` `,` `que` `exista` `persecición` `política` `contra` `el.`

Tokenización basada en caracteres

Usualmente utilizado en investigación de nuevos modelos de NLP. Genera un vocabulario pequeño.

`persona` : `p` `e` `r` `s` `o` `n` `a`

Problemas de la tokenización

- La **tokenización simple y basada en palabras** generan grandes vocabularios de tokens.
- En análisis de texto, gran cantidad de tokens "no representativo" no son considerados (out of vocabulary - OOV). Especialmente los tokens con poca presencia en el texto.
- Este enfoque no permite la identificar semejanzas entre palabras con la misma palabra núcleo: e.g. `persona` y `personas`.
- La tokenización basada en caracteres genera secuencias de tokens muy largas asi como también los tokens individuales son menos significativos.

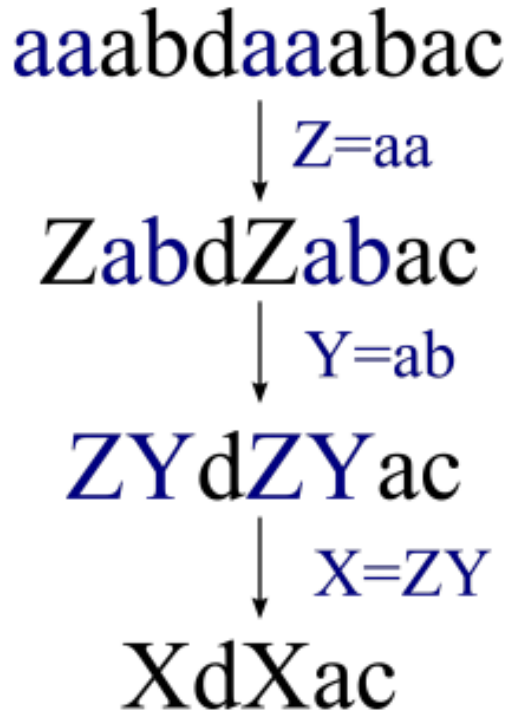
Tokenización basada en Sub-Palabras

- Es una solución intermedia entre la tokenización de caracteres y la de palabras.
- Este enfoque divide palabras poco frecuentes en sub-palabras con mayor frecuencia y significado. por ejemplo:
  - `persona` : `persona`
  - `personas` : `persona` `s`
- Esto ayuda al los modelos de NLP identifiquen mejor el token `persona` con el concepto de **persona**" y el token `s` como el concepto de **pluralidad**.
- Los algoritmos de tokenización mas populares son: WordPiece y Byte-Pair Encoding (BPE).
- Estos modelos son entrenados de forma automática mediante algoritmos.

Byte-Pair Encoding (BPE)

El entrenamiento de BPE se basa en un modelo de compression de datos: "A new Algorithm for Data Compresion", 1994.
Por ejemplo: Comprimir la secuencia "aaabdaaabac"

Solución:
Vocabulario: `X` , `d` , `a` , `c`
donde: X=ZY=(aa)(ab)=aaab
Podemos observar que X almacena mas significado en la secuencia.



Stop words

Stop words es un termino usado en NLP para referirse al conjunto de palabras que no traen información relevante para algunos tipos de análisis. Para cada idioma existe una lista de palabras.

```
'de', 'la', 'que', 'el', 'en', 'y', 'a', 'los', 'del', 'se',
'las', 'por', 'un', 'para', 'con', 'no', 'una', 'su', 'al',
'lo', 'como', 'más', 'pero', 'sus', 'le', 'ya', 'o', 'este',
'sí', 'porque', 'esta', 'entre', 'cuando'...
```

Ejemplos

Tarea 1

```
Ejemplos de Tokenización
```

Tarea 2

```
Ejemplo Word Cloud
```

Part of Speech Tagging

Part of Speech Tagging (POST) explica como una palabra es usada en una sentencia. La anotación puede usar diferentes formatos. Algunas ejemplos son: sustantivos (NOUN), pronombres (PRO), adjetivos (ADJ), verbos (VERB), adposiciones (ADP), conjunciones (CONJ), puntuaciones.

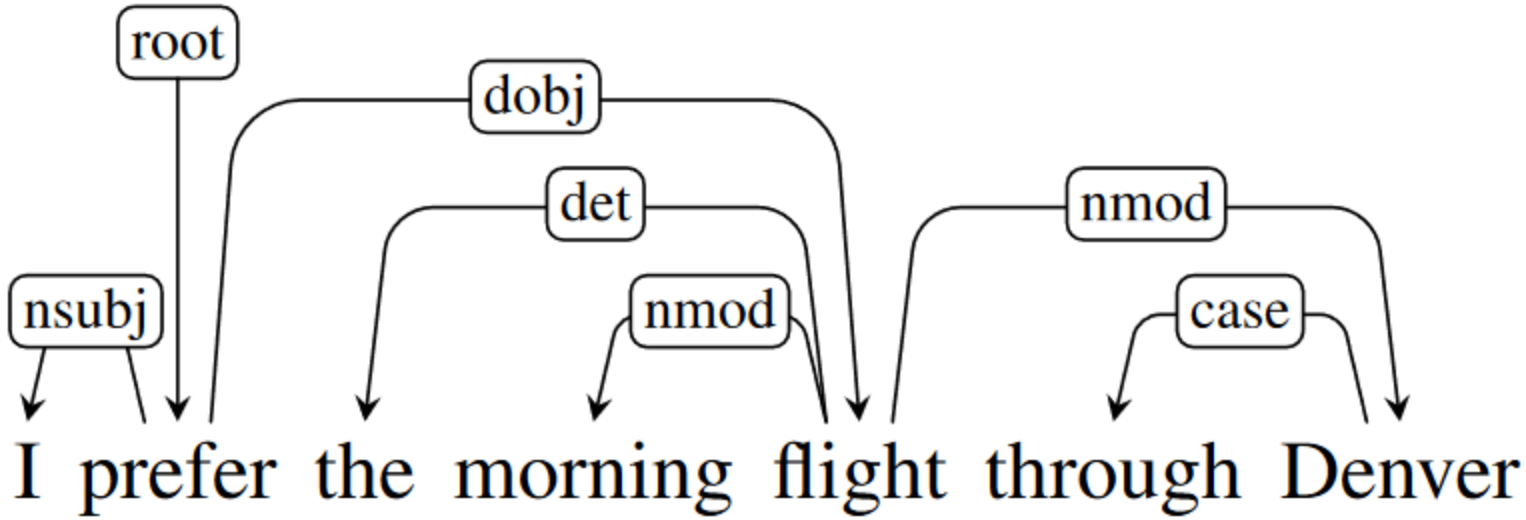
Dada la frase: `El mandatario nacio en Cajamarca`

Solución:

```
word: El          upos: DET   xpos: None  feats: Definite=Def|Gender=Masc|Number=Sing|PronType=Art
word: mandatario  upos: NOUN  xpos: None  feats: Gender=Masc|Number=Sing
word: nacio       upos: VERB  xpos: None  feats: Mood=Ind|Number=Sing|Person=1|Tense=Pres|VerbForm=Fin
word: en          upos: ADP   xpos: None  feats: _
word: Cajamarca   upos: PROPN xpos: None  feats: _
word: .           upos: PUNCT xpos: None  feats: PunctType=Peri
```

Dependencing Parse

Esta estrategia describe la estructura sintáctica en función de **relaciones gramaticales binarias direccionadas** entre palabras.

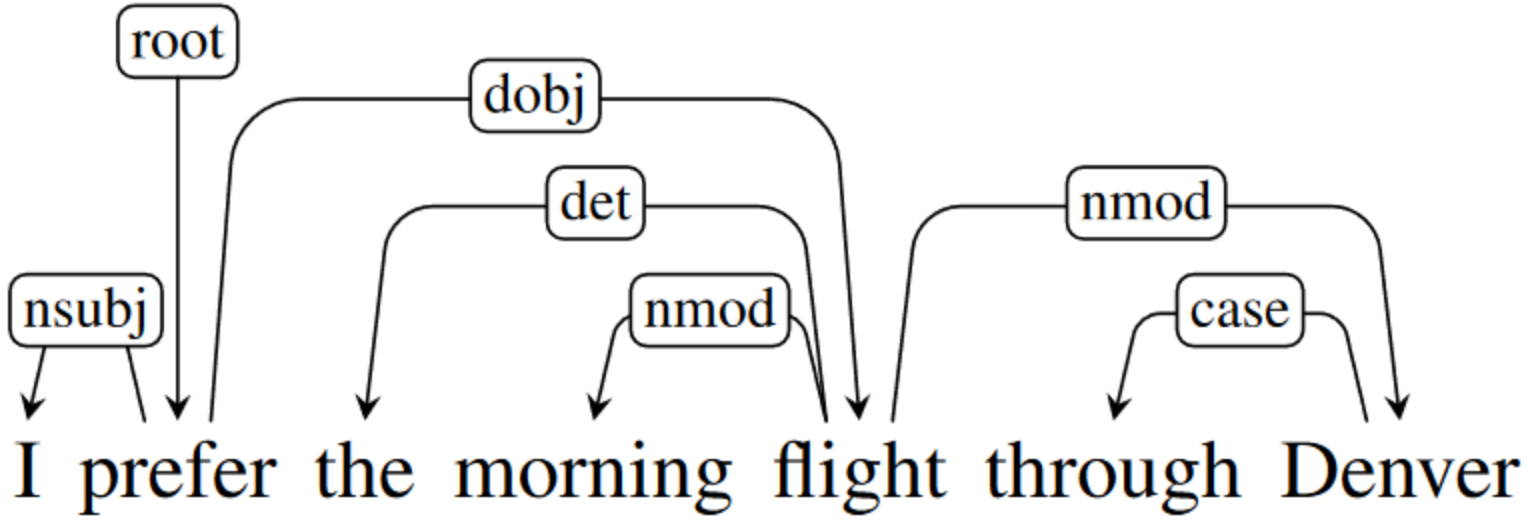


Estas dependencias forman un árbol de relaciones donde **root** indica el inicio de la estructura (La cabeza de toda la estructura).

Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Figure 14.2 Some of the Universal Dependency relations (de Marneffe et al., 2014).

Dependencing Parse



Estas dependencias nos ayudan a clasificar el tipo de relación gramatical que existe respecto al *head*. Se han realizado esfuerzo para crear estándares de dependencias entre múltiples lenguajes. El proyecto **Universal Dependencies (UD)** nos proporciona un inventario de relaciones de dependencias linguísticas aplicable a través de múltiples lenguajes.

Detección de Entidades

La detección de entidades es la tarea de identificar y categorizar información clave en el texto. La entidad puede ser cualquier tipo de palabra o una serie de palabras.
Por ejemplo, reconocer entidades genéricas en el texto:

```
Carlos escucho que la Pontifica Universidad Católica del Perú está en Lima.
```

Solución:

```
entity: Carlos                                  type: PER
entity: Pontifica Universidad Católica del Perú     type: ORG
entity: Lima                                    type: LOC
```

Detección de Relaciones

La detección de relaciones semánticas o eventos ocurre entre dos o más entidades. Por ejemplo, en la frase `París está en Francia`, la parte `está en` indica la relación entre París y Francia. Esta relación es representada por la tripleta: (París,is in, France)

Ejemplos

Tarea 3:

```
Ejemplos NER
```

Bibliografia

- Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, "Introduction to Information Retrieval" (2008)
- Byte Pair Encoding (Paper): http://www.pennelynn.com/Documents/CUJ/HTML/94HTML/19940045.HTM
- Masato Hagiwara , "Real-World Natural Language Processing" (2021)
- Documentación bilblioteca NLTK: https://www.nltk.org/
- Documentación bilblioteca Stanza: https://stanza.com.br/
- Documentación bilblioteca CoreNLP: https://stanfordnlp.github.io/CoreNLP/