Procesamiento del Lenguaje Natural

(y un par de apuntes sobre redes de neuronas)

Víctor M. Darriba Bilbao

Área de Ciencias de la Computación e Inteligencia Artificial

Mayo 2021

Procesamiento del Lenguaje Natural

Resumen:

- Estudio de las interacciones entre humanos y máquinas a través del Lenguaje Natural.
- Mecanismos para la comprensión y generación de lenguaje humano.
- Uso de técnicas de Aprendizaje Automático.
  - **1990s-2010s**: Modelos de Markov, Modelos de Máxima Entropía, Árboles de Decición, SVMs, CRFs,...
  - **2010s-hoy**: Redes de Neuronas Artificiales y Aprendizaje Profundo.

Aplicaciones:

- Traducción Automática
- Sistemas de Diálogo
- Análisis de Sentimiento - Sistemas de Recomendación
- Recuperación/Extracción de Información - Búsqueda de Respuestas
- Síntesis del Habla
- Generación de Resúmenes
- Comprensión/Generación del Lenguaje Natural

Procesamiento del Lenguaje Natural (II)

Tareas comunes:

- Procesamiento de texto y habla: OCR, reconocimiento del habla, segmentación del habla, segmentación de texto (tokenización).
- Morfología: lematización, segmentación morfológica, etiquetación morfosintáctica (desambiguación), stemming.
- Sintáxis: inducción de gramáticas, desambiguación de límite de la oración, análisis sintáctico.
- Semántica:
  - Léxica: semántica léxica, semántica distribucional, reconocimiento de entidades nombradas, análisis de sentimiento, extracción de terminología, desambiguación del sentido de las palabras.
  - Relacional: extracción de relaciones, análisis semántico, etiquetación de roles semánticos.
- Discurso: resolución de coreferencia, análisis de discurso, segmentación y reconocimiento de temas, detección del juício de implicación, minería de argumentos.

Procesamiento del Lenguaje Natural (III)

Frecuentemente, secuencia de tareas:

1. Preprocesamiento y tokenización.
2. Análisis Morfológico.
   - Lematización
   - Etiquetación Morfosintáctica
   - Stemming
3. Análisis sintáctico
   - Parsing Superficial - chunking
   - Análisis de Constituyentes
   - Análisis de Dependencias
4. Análisis Semántico
   - Reconocimiento de entidades
   - Etiquetación de roles semánticos

Etiquetación Morfosintáctica (POS tagging)

Asignar a cada constituyente de la frase (*token*) una etiqueta con:

- Categoría léxica (Part-of-Speech, POS): sustantivo, adjetivo, verbo, ...
- Rasgos morfológicos (opcionales): género, número, persona, caso, ...

Problema: Ambigüedad

Token	$Lema_1$	POS_1	$Lema_2$	POS_2	$Lema_3$	POS_3	$Lema_4$	POS_4
Si	si	**CS**						
trabajo	trabajar	**VMIP1S0**	trabajo	NCMS000				
bajo	bajar	VMIP1S0	bajo	AQ0MS0	bajo	NCMS000	bajo	**SPS00**
presión	presión	**NCFS000**						
bajo	bajar	**VMIP1S0**	bajo	AQ0MS0	bajo	NCMS000	bajo	SPS00
la	la	**DA0FS0**						
atención	atención	**NCFS000**						
.	.	**Fp**						

Etiquetación Morfosintáctica (POS tagging) (II)

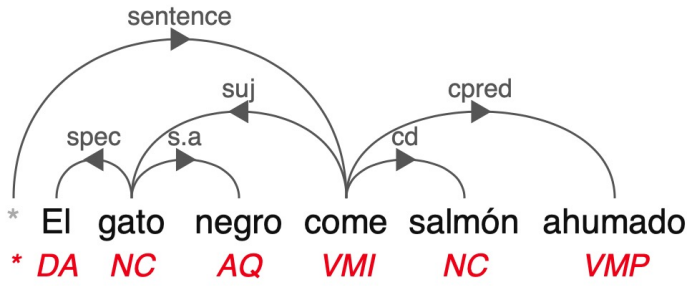
Solución: Desambiguación $\Rightarrow$ etiquetación de secuencias

- Caso particular de predicción estructurada
- Similar a clasificación
- En lugar de predecir una clase para cada *token*, se predicen varias (con pesos/probabilidades)
- Se emplea un algoritmo de búsqueda (recorrido de grafos) para obtener la secuencia de etiquetas más probable para la frase.
- Ejemplo: modelos de Markov, modelos de campo aleatorio condiccional (CRFs)

Datos de entrenamiento: corpora anotados con etiquetas morfosintácticas
Características:

- Aproximación tradicional: *feature engineering*
- Aprendizaje profundo: *embeddings*

Análisis sintáctico de dependencias (Dependency parsing)



1	El	el	DA0MS0	DA	pos=determiner|...	- - -	2	spec	- -
2	gato	gato	NCMS000	NC	pos=noun|...	- - -	4	suj	- -
3	negro	negro	AQ0MS00	AQ	pos=adjective|...	- - -	2	s.a	- -
4	come	comer	VMIP3S0	VMI	pos=verb|...	- - -	0	sentence	- -
5	salmón	salmón	NCMS000	NC	pos=noun|...	- - -	4	cd	- -
6	ahumado	ahumar	VMP00SM	VMP	pos=verb|...	- - -	4	cpred	- -

Fuente: Demo de FreeLing (UPC)

Análisis sintáctico de dependencias (Dependency parsing) (II)

El árbol de análisis se representa como un grafo dirigido:

- Arcos representan relaciones binarias entre palabras o lemas (cabeza y dependiente)
- Cada arco se etiqueta con un tipo predefinido de estructura de dependencia

No precisa de una gramática:

- Las relaciones de dependencia se pueden aprenden
- Banco de árboles: corpus con anotación sintácticas
  - En este caso especificando la posición de cabeza y dependiente y etiqueta del arco

Problema: ambigüedad $\Rightarrow$ muchos árboles son posibles
Solución: predicción estructurada (con grafos en lugar de secuencias)

Reconocimiento de entidades nombradas (NER)

Detectar automáticamente nombres en un texto

- Pueden ser mono o multipalabra
- Determinar el tipo de entidad que representan (persona, lugar, organización, etc)

Solución: etiquetación de secuencias

- Esquema IOB (*Inside, Outside, Beginning*) + tipo de entidad
- Entrenamiento con corpora anotados

Token	POS	NER Tag	Token	POS	NER Tag
Nader	NNP	B-PER	a	DE	O
Jokhadar	NNP	I-PER	well-struck	NN	O
had	VBD	O	header	NN	O
given	VBN	O	in	IN	O
Syria	NNP	B-LOC	the	DT	O
the	DT	O	seventh	JJ	O
lead	NN	O	minute	NN	O
with	IN	O	.	.	O

Evaluación

- Etiquetación morfosintáctica: Exactitud (*Accuracy*)

$$Accuracy = \frac{\#correct\ tags}{\#tokens}$$

- Reconocimiento de Entidades: Precisión (*Precision*), Exahustividad (*Recall*), valor F1 (*F1-score*)

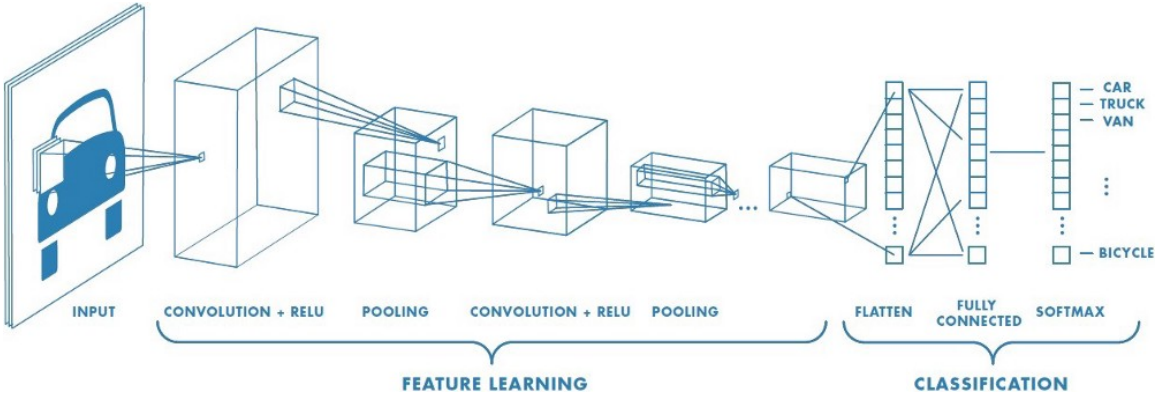
$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN} \qquad F1\_score = 2 \times \frac{Recal \times Precision}{Recall + Precision}$$

- Análisis de dependencias: *Unlabeled Attachment Score* (UAS), *Labeled Attachment Score* (LAS)

$$UAS = \frac{|\{e \mid e \in E_G \cap E_P\}|}{|V|} \qquad LAS = \frac{|\{e \mid l_G(e) = l_P(e), e \in E_G \cap E_P\}|}{|V|}$$

$(e = arco,\ E_G = arbol\ correcto,\ E_P = arbol\ predicho,\ l_X(e) = etiqueta\ arco\ e\ en\ arbol\ X)$

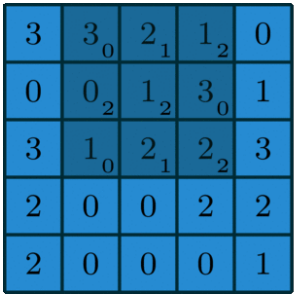
Convolutional Neural Networks (CNNs/ConvNets)



Fuente: towardsdatascience.com

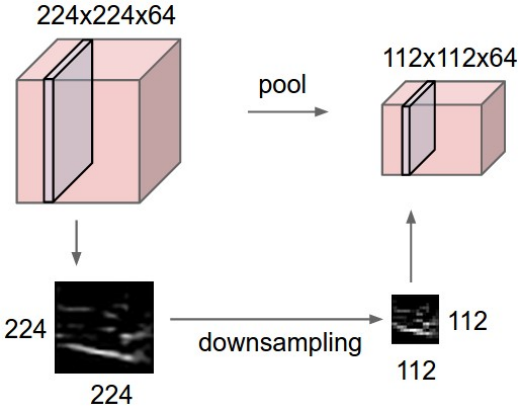
Convolutional Neural Networks (CNNs/ConvNets) (II)

Convolución:



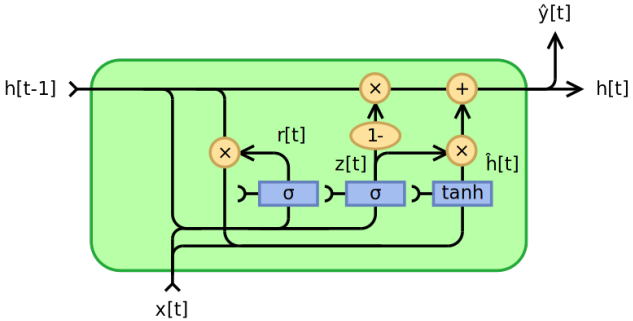
Fuente: towardsdatascience.com

Pooling:



Fuente: Stanford - CS231n

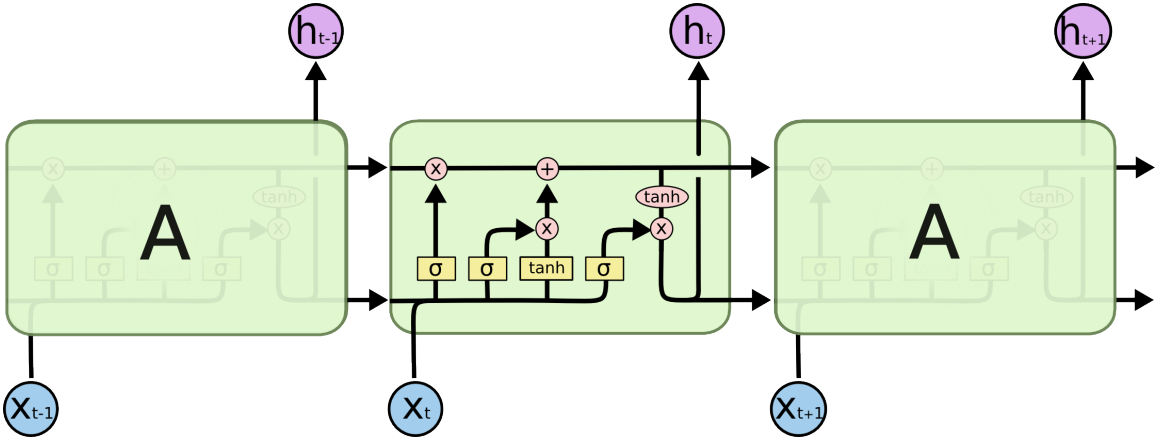
Gated Recurrent Units (GRUs)



Fuente: wikipedia

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r)$$
$$\hat{h}_t = \phi_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t$$

Long Short-Term Memory (LSTMs)



Fuente: https://colah.github.io