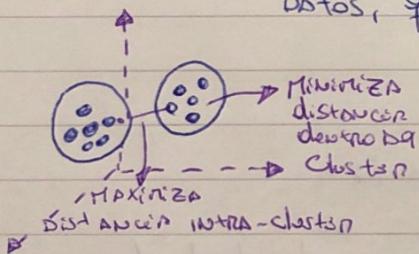


Sup: Encuentran una hipótesis que satisfaga los datos.

↑ Sup: Que el algoritmo encuentre cierta estructura.

### Clustering

- Grupos de instancias (clusters) a partir de info de los datos, que describen objetos y sus relaciones.



Instancias de Cluster:  
- Similares entre sí.  
- Diferentes de otro cluster.

### Típos

#### Partición

- (K-Means)

#### Jerárquicos

- Aglomerativos (bottom-up) De n a 1
- Divisivos (top-down) De 1 a n

## K-Means

IN: - datos no etiquetados ( $x^1, \dots, x^n$ )  $\in \mathbb{R}^n$   
-  $K$  # clusters

1: Inicializa aleatoriamente  $K$  centroides de los clusters.

2: Repetir

- Para cada dato se fija su distancia al centroide y es asignado al mas cercano.
- Mueve los centroides a la posición promedio de los datos

3: Hasta que los centroides no se muevan (o se muevan un % chico)

- Si un cluster (centroide) no tiene datos, se elimina el centroide

#Cluster < # DATOS



Init. Centroides

- AL AZAR

\ elegir  $K$  datos cualesquiera

- Pueden converger a distintas soluciones dependiendo de como iniciaza

Multiples inicializaciones AL AZAR (evitar óptimas locales)

- elijo cluster con menor fuerza de corte

(util en casos chicos  $K < 10$ )

K-Means intenta minimizar

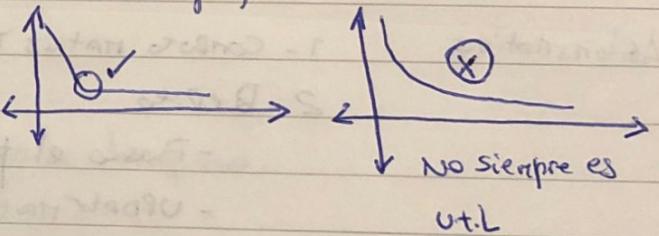
- Asignación d/cluster: minimizar Asigna los puntos al centroide más cercano
- Movida de centroides

<b>DISTANCIAS</b> Eucl. dep : $\sqrt{\sum_{i=1}^m (x_i - y_i)^2}$ Manhattan : $\sum_{i=1}^m  x_i - y_i $ Chebychev : $\max_{i=1 \dots m}  x_i - y_i $	<b>OTRAS :</b> - Similitud de Cadenas - Jaccard (Documentos) - Hamming & Levenshtein (Cadenas de caracteres)
--	--

D-Levenshtein: Minimo numero de ediciones de caracteres (Poner, borrar, cambiar)  
 Referencias para comprobar una palabra por otra. uso en ADN  
 $D_1(\text{Honda}, \text{Hyundai}) = 3$

Eleccion del K : - MANUALMENTE (puede ser ambiguo)

- Elbow Method



- EVALUAR CON UNA MATRIZ Y USAR CON BON SENSE PARA UN PROPOSITO POSTERIOR.

Ventajas

- Algoritmos Simples
- Eficientes

Desventajas

- Sensible a la elección de los K centroides iniciales
- Sensible al Ruido y Outliers
- Hay que especificar el K.
- Problemas para detectar clusters naturales cuando no tienen forma esférica o tanques a formas muy diferentes.

## Expectation Maximization & Méjica De Gaussiana

- Assumir overlap (hay una probabilidad de que los datos estén en más clusters)
- Cada cluster corresponde a una distribución de probabilidad (normal o Gaussiana). Se quiere conocer MEDIDA & VARIANZA
- A diferencia de K-Means comprobamos qué cluster es un elemento visto en distintos clusters
- USOS: Reconocimiento Del hablante.

## Jerárquico

### Abormentivo

1 - Construir matriz De Distancias

2 - Repito

- Busco el par de puntos mas cercanos + Merge
- Update matriz De Distancias

3 - Hasta que haya un solo cluster

### Definiendo Distancia entre clusters

- MIN (Single Link): distancia mínima entre los puntos de dos clusters
- MAX (Complete link): distancia máxima entre los puntos distintos de dos clusters
- AVG: promedio de la distancia entre los puntos de los clusters
- Distancia entre centroides.

Método De Densidad

	$p_1$	$p_2$	$p_3$
$p_1$	0	x	y
$p_2$	x	0	z
$p_3$	y	z	0

- Sistematico

- Cálculo en la dimensional

### Ventajas

- No tiene requerimientos de clústeres (se pueden obtener agrupando el resultado en el nivel deseado)
- Pueden componerse con taxonomías jerárquicas (Reino Animal)

- Se puede usar K-near points, después largos.

### Desventajas

- Sensible al ruido y outliers
- Computacionalmente más costoso (tanto a disposición que K-means)

### DB SCAN

- Basa requiere de alto desgaste que estos separados de otra por espacios de larga distancia.

① Core-Points: Puntos en el interior de una región densa (Mínimo de n vecinos Mpts).

② Border-Points: Vecinos de algún core-points

③ Noise-Points: No son Core ni border.

- Algoritmo:
1. Etiquetar todos los puntos como border || noise || core
  2. Eliminar puntos noise
  3. Poner un eje entre todos los core-points que estan en UNDENTIFIED. Distancia especifica (EPS)
  4. Crear clusters con los grupos de core-points conectados
  5. Asignar cada border-point a algun cluster

### Ventajas

- No hay que especificar K
- Encuentra clusters de manera Arbitraria (en forma y tamaño)
- Robusto Al Ruido
- Puede encontrar clusters que K-Means no puede

### Desventajas

- Elegir EPS & MinPts requiere conocimiento de los datos: Difícil en casos de alta dimensionalidad
- Tiene problemas cuando hay mucha variedad de dimensionalidad
- Es costoso en Alta Dimensionalidad

### Evaluación De Clusters

- ↳ Determinar la tendencia del Clustering, en un set de datos.
  - ↳ Determinar el correcto numero de clusters.
  - ↳ Evaluar el resultado de un análisis de cluster, que tan bien "fit" los datos sin referencia a info externa
  - ↳ Evaluar el resultado con info externa.
  - ↳ Comparar dos set de clusters para determinar cual es mejor
- } No necesitan info externa
- } Necesitan info externa
- } Superiores o NO

## Medidas

→ Internas (Unsupervised): - No usan info externa.

- / SSE
  - / Cohesion intra clusters
  - / Separacion entre clusters
- Que tan distintos & bien separados estan los clusters.

→ Externas (Supervised): / Revision que tanto matches info externa

- / Usan informacion no presente en el DataSet
- / Entropia
- / Representar imagenes, identificar || eliminar outliers.

→ Relativas: / Comparan distintos clusters & clusterings (pueden ser supervisado & no).

- / Ejemplo: Distintos numeros de clusters.
- / K-means se puede comparar usando SSE & Entropia

## Coeficientes De Silhouettes

- Combinan cohesion & separacion

- Varia de -1 a 1. un valor negativo no es deseado. queremos que sea positivo y tan cercano a 1 como sea posible.

## Matriz De Similitud

1. Armar matriz ( $S_n$ ): valor de similitud entre 0 y 1

2. Ordenar filas y columnas de acuerdo a los elementos de los clusters.

3. Inspeccionar visualmente (diagonal por bloques) o utilizando correlacion con Ground Truth.

## Métodos externos

- Criterios
- Homogeneidad
  - Completitud: Objetos de la misma categoría asignados al mismo cluster.
  - Mejor un nuevo cluster/categoría que nacer cluster con objetos heterogéneos.
  - preservar cluster pequeños

- Medidas
- basadas en Matching (pureza, F-measure)
  - basadas en entropías
  - Información de los pares
  - Medidas de correlación

Si tenemos info externa para clusters y obtenemos una medida hará parte de la distribución estadística, para ver qué tan probable es que esto suceda por casualidad.

## ENSAMBLES DE MODELOS

- Intentan mejorar el rendimiento de los modelos de ML.

Error debido a:

- Sesgo/Bias : Diferencia entre predicción del modelo (o promedio de ellas) y valor correcto
- Variancia : La variabilidad de la predicción del modelo para unos datos dados.  
Cuanto varían las predicciones para un set de datos entre distintas realizaciones del modelo

$$y = f(x) + \epsilon$$

,  $y$  = lo que queremos predecir  
 $x$  = Coordenadas

$$\text{Err}(x) = \text{Bias}^2 + \text{Varianza} + \text{Error Irreducible}$$

Disminuir el error  $\rightarrow$  Método con bajo sesgo/bias y bajo variación

Algoritmos de Aprendizaje ~~instables~~: sufren cambios importantes ante pequeñas variaciones en datos de entrenamiento  
(Árboles de decisión, Redes Neuronales)

→ ~~Estables~~ : Regresión Lineal, vecino más cercano.

Predictores :

- Rígidos : Poca flexibilidad (menos complejos). Mayor error de sesgo (estimación de un error muy complejo con un modelo simple p.ej: Lineal)
- Flexibles : MÁS complejos. MAYOR error de variación y MENOR de sesgo.

Dilema Bias-Variancia: A bajo sesgo/Bias Alto Variancia  
 A Alta Variancia bajo Sesgo

• bajo Sesgo/Bias + Alta Variancia: curva fue pas por cada ejemplar de ENTRENAMIENTO

• alta Variancia + Alto sesgo/Bias: Lineas Rectas que atraviesa los datos

SOLUCIONES:

→ USANDO predictores con bajo sesgo disminuir la Var.  
 (muchos predictores y promediando; bagging a Random Forest)

→ Reducir Sesgo de Predictores

(construir predictores en series de forma de disminuir el sesgo, boosting)

USO DE UN CONJUNTO DE MODELOS PARA CONSTRUIR UN META-MODELO  
 COMBINANDO DECISIONES (Votación simple, votación ponderada, promedio, etc)

ENSAMBLE	PLANOS	DIVISIVOS
	Comité de expertos en un mismo tema con f Opinión en algunos casos	Comité de expertos en f áreas Se necesita saber como combinar los resultados
ALGORITMOS	BAGGING BOOSTING RANDOM FOREST	STACKING MEZCLA DE EXPERTOS

## • Bootstrapping

- Herramienta ESTADÍSTICA
- TÉCNICA DE RESAMPLEO A PARTIR DE UN CONJUNTO DE DATOS CON REEMPLAZO.
- SE PUEDE USAR PARA MEJORAR MODELOS DE APRENDIZAJE (ARBOLINES)

## • Bbagging (Bootstrap Aggregating)

- REDUCIR VARIANZA DE UN MODELO
- PROMEDIANO CJO. DE OBSERVACIONES REDUCE LA VARIANZA (TOMA MUCHOS CONJUNTOS DE ENTRENAMIENTO Y HACE PROMEDIO DE LAS PREDICCIONES)
  - ↳ COMO NO TENGAS MUCHOS CONJUNTOS DE ENTRENAMIENTO TENDRÁS REPETIDAS DEL MISMO PUEDE TENER

PREDICCIÓN → NUMÉRICA: PROMEDIO  
CATÉGORICA: VOTACIÓN

- SE GENERAN 3 DATASETS DE ENTRENAMIENTO CON BOOTSTRAP.
- SE ENTRENA EL MÉTODO CON EL  $b$ -ESIMO CONJUNTO DE ENTRENAMIENTO Y OBTIENE LA PREDICCIÓN EN EL PUNTO  $x$ .
- SE PROMEDIAN TODAS LAS PREDICCIONES.
- ARQUITECTURA PARALELA
- ÚTIL PARA RESULTADOS SENSIBLES A LOS CJOS DE ENTRENAMIENTO.

## • Random Forest

- MEJORA → BAGGING CUANDO SE USA CON ARBOLINES  
(SI HAY ATRIBUTOS QUE SON PREDICTORES FUENTES (ALTO INFO-GAIN) LOS ARBOLES SERÁN MUY SIMILARES)
- SE INTENTA ELIMINAR CORRELACIÓN DE LOS ARBOLINES, PARA REDUCIR VARIES MEJOR PROMEDIAR CANTIDADES NO CORRELACIONADAS.
- EN CADA SPLIT DE UN NODO SE CONSIDERA SOLO UN SUBCONJUNTO  $M$  DE LOS  $P$  ATRIBUTOS ( $M$  ELEGIDOS AL AZAR)  $M = \sqrt{P}$

## Boosting

• Difiere de bagging en que cada modelo se arma usando información de modelos anteriores. Se computan pesos para mejorar los errores en cada algoritmo DIO MAL.

Procedimiento:

- / Los modelos simple entrena sobre todos los datos
- / En cada iteración  $i$ , entrena  $h_i$  dando mayor importancia a los datos mal clasificados por la iteración anterior (DAND distintos pesos)

- / Terminar Luego de un número de iteraciones.
- / Clasifican nuevas instancias usando una votación de todos los modelos construidos

- Weight-based boosting (ADA Boost)

## Stacking

• Hacer varias predicciones a un Hold-out Data Set

• Armar esas predicciones para armar un nuevo Data Set para entrenar un nuevo modelo.

- ① Dividir en entrenamiento y validación
- ② Entrenar  $f$  modelos con el Dataset de entrenamiento
- ③ Hacer predicciones con los modelos entrenados sobre Hold-out
- ④ Usar resultados de (3) para entrenar otro modelo (Meta-Modelo)

- / Arquitecturas Paralelas (Bagging) vs Secuenciales (boosting)
- / Nuevo clasificador (lo responde) vs  $f$  clasificador (stacking)
- /  $f$  formas de combinar resultados.

- / Para clasificadores Instables (árboles) usar bagging & Random Forest
- ||   ||   Estables (NBAYES) usar boosting
- Sobre todo   • Disminuir var usando predictores de bajo sesgo (bagging & Random Forest)
- Dileno   • Reducir sesgo de predictores (boosting)

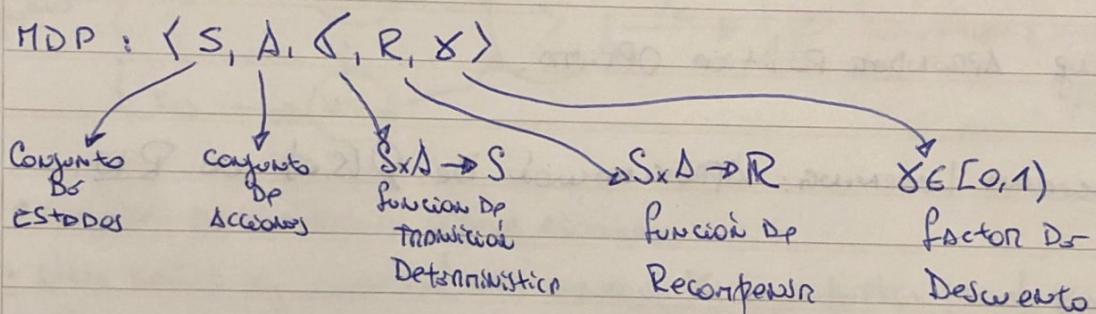
## Aprendizaje Por Refuerzos

Agente Autónomo: Percebe e interactúa con el Ambiente (Elige acciones óptimas para lograr objetivos)

- / NO tiene entiendido, interactúa con el Ambiente y tiene premios y castigos
- / Tarea: Aprende a elegir acciones óptimas para lograr su objetivo

Dado un estado inicial el agente elige acciones que maximizan la recompensa acumulada en el tiempo.

## MDP (Markov Decision Problem)



La recompensa y la transición de un estado a otro dependen solo del Estado Actual (no anteriores).

- El Agente percibe el estado  $S_t$ , elige y realiza una acción  $Q_t$ .
- El Ambiente Responde dando una Recompensa  $R_t = R(S_t, Q_t)$
- $R(S_t, Q_t)$  depende solo del estado Actual, no de los anteriores.

Se busca política que arroje MAYOR RECOMPENSA ACUMULADA.

$\sum_{t=0}^{\infty} \gamma^t r_t =$  Función De Valor: Recompensa Acumulada al seguir una política  $\pi$  para seleccionar estados a partir del estado  $S$ .

Función Objetivo: Política de control que MAXIMIZA la Función de Valor.

$$\text{Fórmula de Value} = \sqrt{\pi}(s)$$

$\pi$  = Política Optima

$Q(s, a) = \text{máxima ganancia esperada desde } s \text{ ejecutando } a$

### Dilema explotación - EXPLORACIÓN

#### • Estrategia $\epsilon$ -First

- Prob 1- $\epsilon$  se elige al azar (EXPLORACIÓN)
- Prob  $\epsilon$  se elige mejor acción conocida (EXPLOTACIÓN)

#### • Estrategia $\epsilon$ -greedy

- Prob  $\epsilon$  se elige al azar
- Prob 1- $\epsilon$  se elige mejor acción conocida.

$Q$ -Learning: Aprender Política Optima.

Deep Reinforcement Learning: Aproximación De  $Q(s, a)$  con RN.

### Características

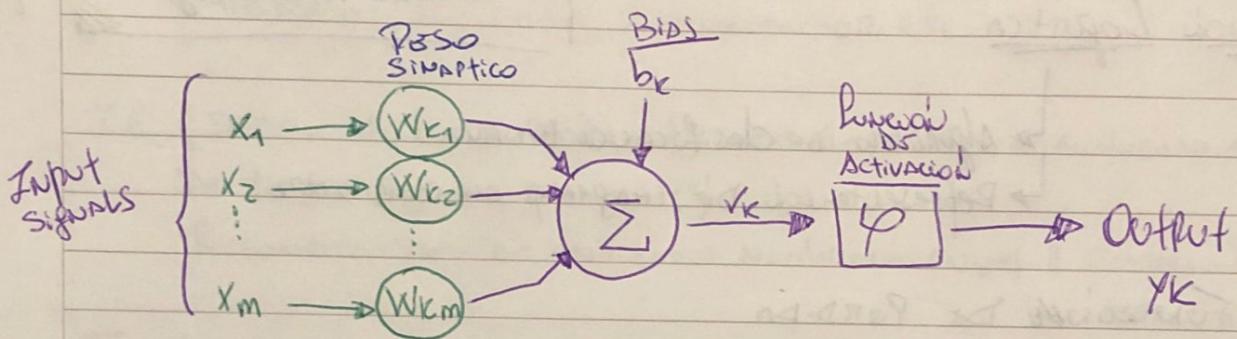
- ↳ Aprendizaje Permanente
- ↳ estados parcial o totalmente observable
- ↳ Recompensa tardía,
- ↳ estados finos de experimentación

## REDES NEURONALES

APRENDIZAJE SUPERVISADO:

- Perceptrón Simple
- Redes feedforward multicapa

- Basado en modelos biológicos
- Opera en Paralelo
- Roberto Atoñ Follas

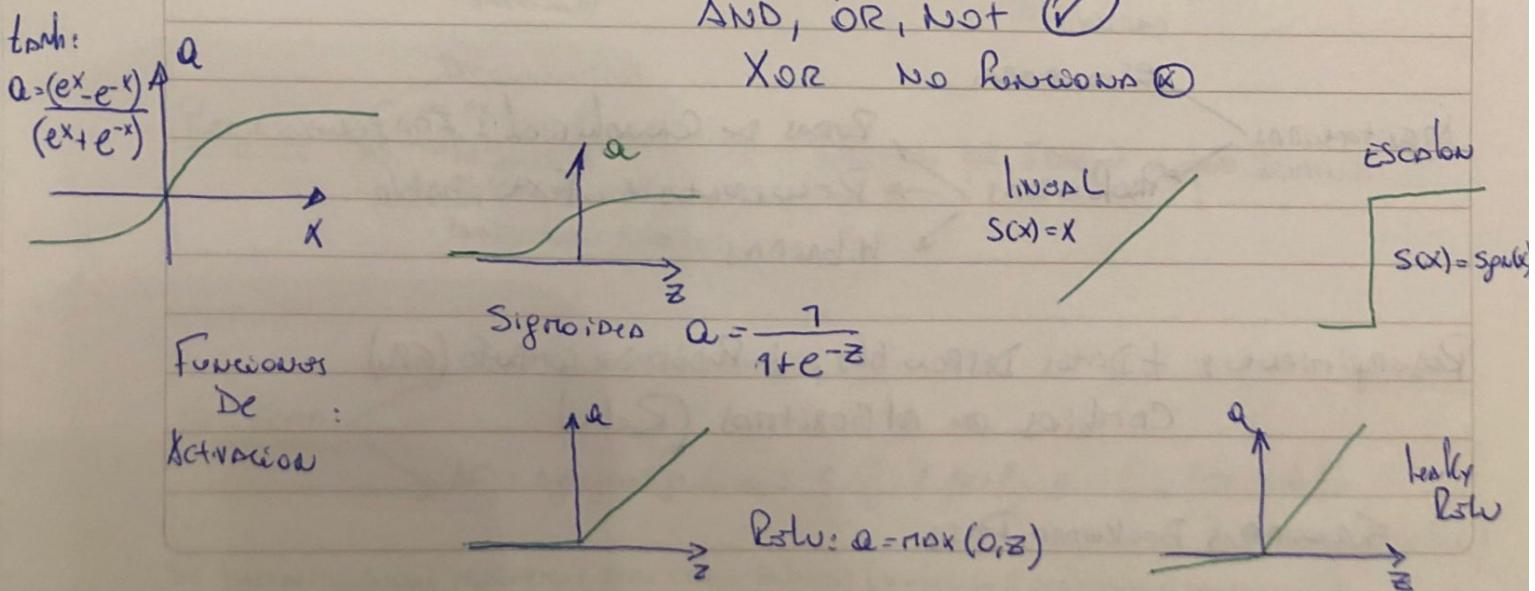


- Conjunto de conexiones y pesos Asociados
- Una señal  $x_j$  conectada a la neurona  $k$  se multiplican por el peso  $w_{kj}$ .
- Sumador de señales  $\rightarrow$  Arreglo comb. lineal de las entradas
- Función de Activación limita la amplitud de la salida.
- $b_k$  variable actividad de la neurona

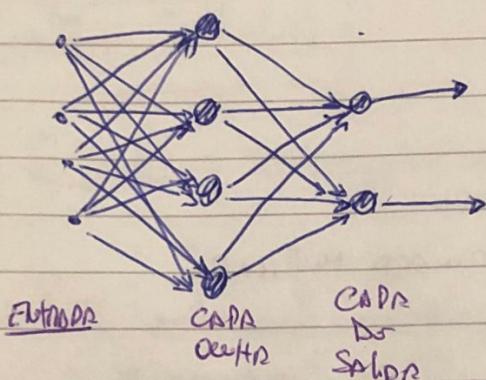
Perceptrón (Simple): Resuelve problemas linealmente separables

AND, OR, NOT  $\bigcirc$

XOR No Resolvió  $\times$



## Def: Feed forward



## Arquitectura

- Perceptron multicapa
- Conexión total

### Parámetros de PN

- f de activación
- pesos iniciales
- # CAPAS OCULTAS
- CONEXIÓN ENTRE CAPAS
- TASA D $\rightarrow$  APRENDIZAJE

Max f<sub>g</sub>  
EVASIVAS  
EN  
CAPA  
OCULTA  
Z

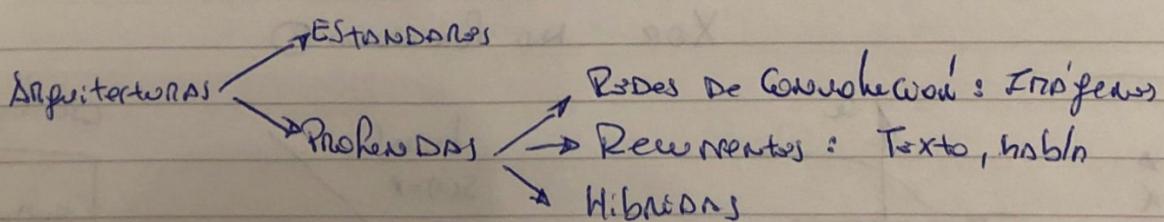
## Regresión Logística

- Algoritmo de clasificación binaria
- Representación de imágenes en binario

- Funciones de Perdida
- Función de Costo

## Algoritmo de Descenso por Gradientes

- Encuentran  $w, b$  que minimizan  $J(w, b)$
- Variantes para acelerar la convergencia.



Requerimientos: + Datos Disponibles, + Poder de Cálculo (GPU)  
Cambios en Algoritmos (ReLU)

FORWARD y BACKWARD PASS

## Text-Mining

El texto puede ser informal (twitter, informes médicos, etc) con errores ortográficos, abreviaturas, typos & formal (artículos científicos, periodísticos) bien formados

NLP (Procesamiento De lenguaje NATURAL): Técnica computacional de procesar lenguaje Natural para analizarlo y/o generar lo

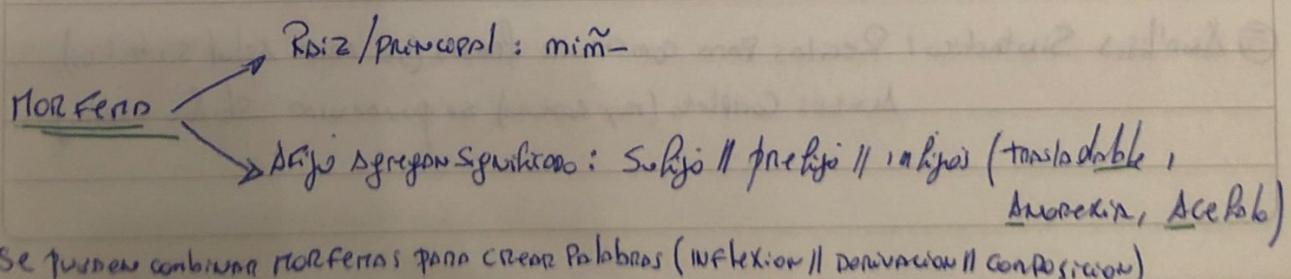
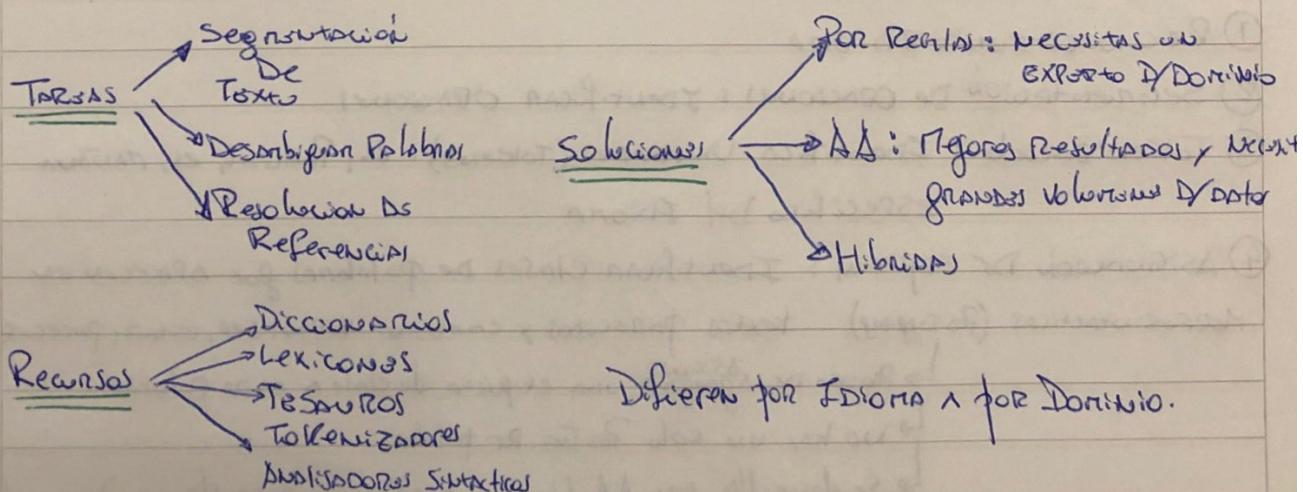
- Necesita conocer del lenguaje (Sintaxis, Morfología, Semántica, etc).
- Traductor, Resumidor, Autocorrector, etc.

IE (EXTRACCIÓN DE INFORMACIÓN): extrae info estructurada a partir de textos ("llenar plantillas").

- Reconocimiento De ENTIDADES Nombradas (NER) || EXTRACCIÓN De Relaciones (RE).

TM (MINERÍA DE TEXTOS): Analiza la info para descubrir patrones & conocimientos no mencionados explícitamente.

- OPINION Mining, detección SPAM, Análisis De Sentimientos.



LM (Modelo de Lenguaje): modelo que asignan probabilidades a secuencias de palabras.

Se usan N-gramas para estimar la proba de una palabra dado los anteriores.

Representación  
Palabras

• One-Hot-Vector: CADA Palabra constituye una posición en el vector con "n" como #Apariciones.

- Mucho vocabulario  $\Rightarrow$  Vector muy GRANDE
- NO Representa similaridad entre palabras

• Word-Embedding: CADA palabra es un Representante por un vector denso (s.w. cerca).

- Representa palabras por su contexto
- APRENDE: Word2Vec, GloVe, etc.

Stemming: Reduce alijo para obtener la Raiz (RUNNING: RUN, consider: consider)

Lematización: obt. en sentido de diccionario de una palabra "Lema" (DID: DO) want: go

## // ARQUITECTURA IE // NLP //

① Reconocimiento Idioma

② Segmentación De Oraciones: Identifican ORACIONES

③ Tokenización: Identifica UNIDADES (Tokens): EL, Parcial, es, MAÑANA específico Del Idioma

④ Asignación De Etiquetas: Identifican CLASES DE PALABRAS que APARECEN EN MORFO-SINTACTICAS (Pos Tagging) textos pertenecientes y cambian: VERBO: SER, ESTAR, PARECER.

↳ Proceso de <sup>ASIGNACIÓN</sup> Agregar una etiqueta de clase a CADA Palabra

↳ No hay un solo Postag por palabra

↳ Se desarrolló con AA (buena performance para algunas lenguas)

⑤ Análisis Sintáctico: Reglas para construir frases (se arma Árbol Sintáctico).

Análisis Completo (muy costoso) se puede usar Shallow-Parsing

## ⑥ Análisis Semántico: "El gato come polillas"

E gato(x) AND J come(y) AND come(x,y)

## ⑦ Reconocimiento De

Entidades Nombradas (NER): Identifican instancias de una clase de información específica en el texto y asignarlos una clase (Personas, empresas, Lugar, etc)

- ↳ Por Reglas, usando AA & Hibrido
- ↳ Consideración total // Parcial

## ⑧ Extracción

De  
Relaciones : Detectar un tipo de Relación específico entre entidades nombradas (Medicamento, Relación, ubicación)

Possible Paso)

Segmentar  $\Rightarrow$  Pos Tagging  $\Rightarrow$  Extraer  $\Rightarrow$  Dividir  $\Rightarrow$  Entrenar  $\Rightarrow$  Reportar  
Features      Datos      Modelos      Resultados

108