1. Introducción

Historia de la PLN

1940-50s: **Autómatas** (Chomsky), **Modelos probabilísticos** (Shannon)
60-70s: **Simbólico** (Lenguaje formal, inteligencia artificial), **Estocástico** (método bayesiano)
70-80: HMM, XXX Grammar, Redes semánticas, Discurso
80-90s: empirismo (modelos probabilísticos), Modelos basado en los datos
90-00s: primeros sistemas comerciales
2000>: LDC, Aprendizaje no supervisado, Deep Learning

Aplicaciones del PLN

- Agentes conversacionales
- Traducción automática
- Búsqueda de respuestas

Conocimientos clave de PLN

Fonética y fonología: los sonidos
Morfología: componentes/variaciones de palabras
Sintaxis: el orden de palabras
Semántica: el significado de palabras
Pragmática: la intención de una frase
Discurso: la historia de una conversación

Donde entra el contexto? En la semántica?

2. Análisis morfológico

Que son los elementos principales de la morfología?

La morfología estudia como descomponer las palabras en elementos indivisibles.

morfema: la unidad mínima de una palabra que expresa un significado
lema: el morfema raíz de una palabra (mujeres -> **mujer** +s)
lexema: un morfema que modifica el lema (mujeres -> mujer **+s**)
lexicón: un diccionario de todos los morfemas y como clave su tipo

reglas ortográficas: describe los cambios de ortografía que ocurren cuando se combinan dos morfemas
hechos morfotácticos: describe el orden en cual se juntan morfemas (características morfológicas)
características morfológicas: la información gramatical de los distintos morfemas de un análisis (+N, +Masc, +pl)

El análisis morfológico necesita: **reglas ortográficas + Hechos morfotácticos + Lexicón**

Como se utilizan autómatas finitos para el análisis morfológico?

Los autómatas se pueden utilizan para detectar si una palabra sigue una estructura concreta y para aplicar cambios como hechos morfotácticos y reglas ortográficas.
Además, se utiliza para hacer un análisis morfológico, dado una palabra en su forma '**superficial'**, es capaz generar su forma **léxical**.

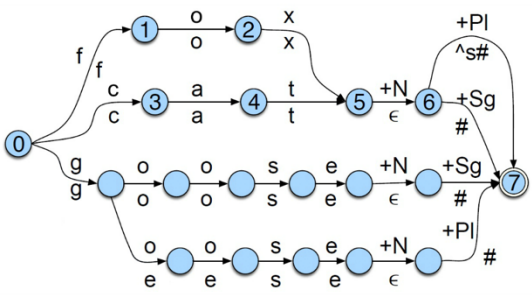
Nivel superficie -> nivel léxico (Vinos -> vino +N +Masc +Pl)

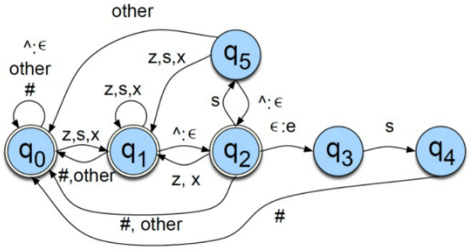
En este autómata finito, los arcos tienen dos propiedades (arriba y abajo) (autómata con una cinta con dos símbolos)

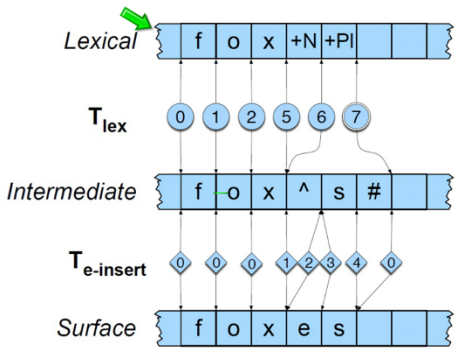
- Arriba indica el símbolo que se genera en la salida
- Abajo indica que símbolo es necesario en la entrada

Símbolos especiales
∈ ... vacío
... frontera entre palabras/espacio
^ … frontera entre morfemas

- Para ser capaz de aplicar reglas ortográficas (de cuales hay muchas) se separa el análisis morfológico en dos pasos:
- Aplicandolas revez, se generan los palabras en function de su forma lexical.
- Los TSF (Transductores de estado finito) no pueden resolver ambigüedad (vino: bebida o venir)







3. ETIQUETADO MORFOSINTÁCTICO

Morfosintaxis = Morfología + Sintaxis

La M. sirve para identificar los diferentes partes de una oración (part-of-speech: POS tagging)

Categorías morfosintácticas son: Substantivo, Determinante, Adjetivo, Pronombre, Verbo, Adverbio, Proposición, Conjunción, Interjección, …

Hay algunos estándares de tags (Treebank, EAGLES, …).

Como se hace el POS tagging?

Un HMM se entrena utilizando un corpus de frases, calculando

- Probabilidad de **transición**: la probabilidad de un tag sigue otro, p(T_2 | T_1)
- Probabilidad de **emisión**: la pro, de la aparición de una palabra data un tag. p('eat' | T_1)

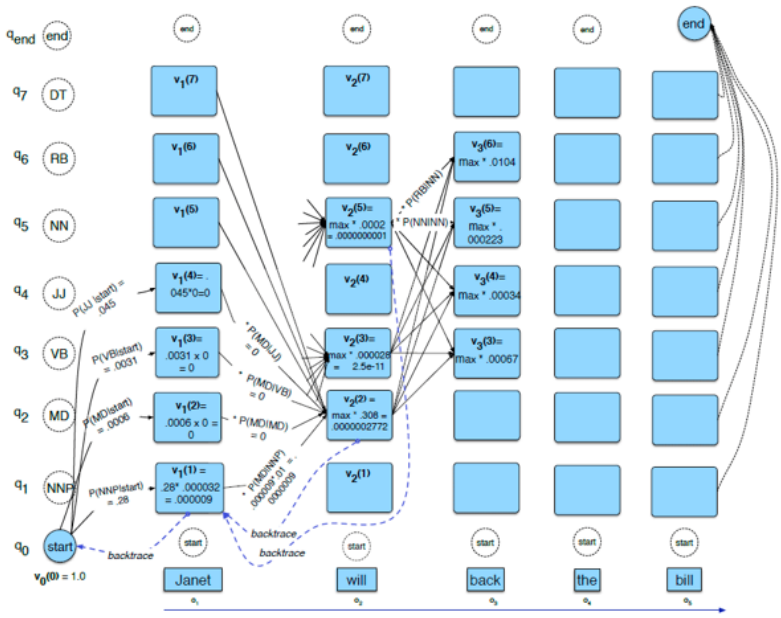
Con este modelo se pueden encontrar los tags que son las más probables para las palabras de una oración.
Es capaz de resolver ambigüedad basado en las probabilidades de secuencias de tags.

Que es el algoritmo viterbi?

Viterbi es una búsqueda por amplitud exhaustiva para encontrar los pos tags más probable de una oración.

$$V_{i+1} = V_i * p_{transición} * p_{emisión}(w_{i+1})$$

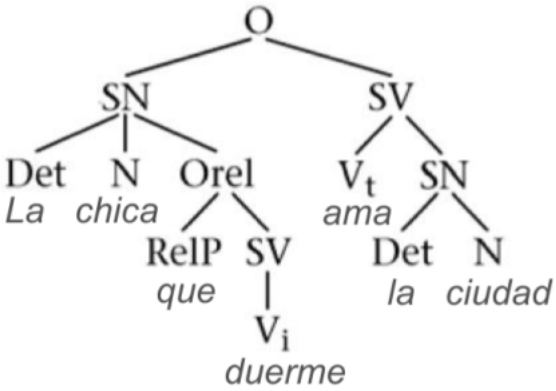
Se elija el camino con la mayor probabilidad de start hasta end.



4. GRAMÁTICAS PARA EL ANÁLISIS SINTÁCTICO

Que es el análisis sintáctico?

Se asigna un **sintagma** a cada palabra de una oración, generando un **árbol sintáctico**.



Utilizando estos sintagmas se pueden definir una **gramática sintagmática.**

1) Vocabulario Terminal: T
2) Vocabulario no terminal: N
3) Reglas: R
4) Símbolo Inicial: S

Se pueden utilizar **descendente** (gramática->texto), o **ascendente** (texto -> gramática)

Un **constituyente sintáctico** es una palabra o una secuencia de palabras que realizan una función conjunta dentro de la estructura jerárquica de la oración.

La potencia de la gramática (su capacidad de definir un lenguaje) es una función de las reglas que tiene.

Hay 4 niveles de Reglas (Chomsky)

- Tipo 3: Estado finito A→x, A → Bx
- Tipo 2: **Libre de contexto**: A→x, A → Bx, A→, A → AYZ
- Tipo 1: Sensible al contexto: A→x, AX → Bx, AX → Axz
- Tipo 0: enumerable recrusivamente:

Para NLP la Libre de contexto es la más importante, porque se puede utilizar para lenguajes naturales.

Sintagma vs. POS tags

- Sintagmas reflejan una relación entre palabras; POS tags sirven para entender que tipo de palabra es.
- Las Sintagmas se aplican en función de Gramáticas, las POS tags dependen de su distribución en el corpus.
- Aplicando los sintagmas no hace falta tener un corpus!
- Sirven como base para el análisis semántico.

G. con categorías complejas / de Unificación

- aumentar la capacidad de la gramática con categorías que no sean atómicas. Las reglas tienen ser capaz de juntar las propiedades de estas categorías complejas.
- Aumentan la expresividad, recogiendo la información de concordancia. Por ejemplo, concordancia de género y número.

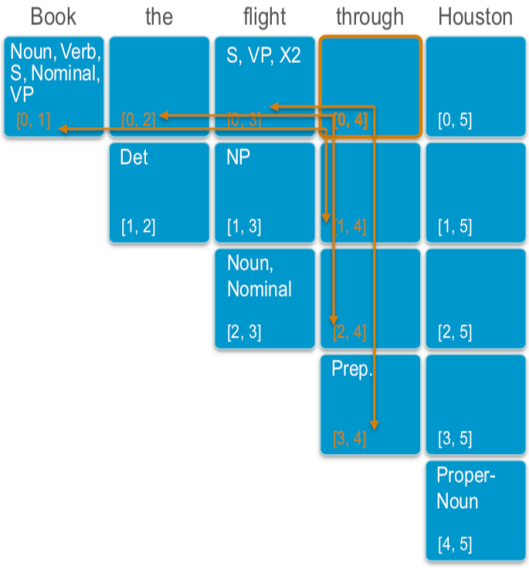
5. ANÁLISIS SINTÁCTICO

Utilizando una gramática sintáctica y **la programación dinámica**, se pueden generar árboles sintácticos.
Hay ambigüedad de estos árboles, porque hay múltiples para la misma oración.
Se pueden resolver estas ambigüedades utilizando métodos probabilísticos.

Programación dinámica: la solución óptima de subproblemas, permite encontrar la solución óptima del problema entero.

Cocke-Kasami-Younger (CKY)

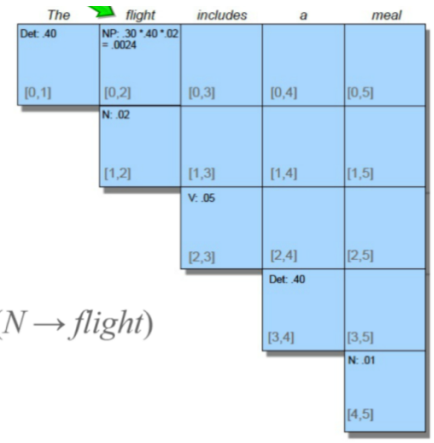
Es un algoritmo de programación dinámica, utilizando una gramática libre de contexto en formato ***Chomsky Normal Form*** **(CNF)** para generar el árbol sintáctico de una frase.



Algoritmo CKY probabilístico

Es una extensión, para resolver ambigüedad, en cual cada regla lleva una probabilidad y se calcula el árbol más probable.

S	$\rightarrow NP\ VP$	.80	Det	$\rightarrow the$	.40
NP	$\rightarrow Det\ N$	.30	Det	$\rightarrow a$	.40
VP	$\rightarrow V\ NP$	.20	N	$\rightarrow meal$	.01
V	$\rightarrow includes$	.05	N	$\rightarrow flight$	.02



Celda [0, 2]

$$P(\text{NP}) = P(NP \rightarrow Det\ N) \cdot P(Det \rightarrow the) \cdot P(N \rightarrow flight)$$
$$= 0.30 \cdot 0.40 \cdot 0.02 = 0.0024$$

6. SEMÁNTICA Y REPRESENTACIÓN DEL SIGNIFICADO

Que es la semántica, que tipos hay, y como representarla?

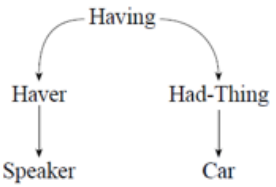
La semántica es el significado y hay dos tipos

- **Semántica composicional**: significado de un sintagma y la oración en función de las palabras de la oración
- **Semántica léxica**: significado basado en definiciones de un lexicón o cosas similares

Maneras de representación:

- **Frames**: Sistemas estructuradas con variables/vuecos
- **Lógica descriptiva**: Subconjunto de lógica del primer orden (optimizada)
- **Diagrame de dependencia conceptual**:
- **Redes semánticas**: Define la semántica en función de las palabras alrededor y su relación

$$\exists e,y\ Having(e) \wedge Haver(e,Speaker) \wedge HadThing(e,y) \wedge Car(y)$$



Car
⇑ POSS-BY
Speaker

```
Having
 Haver:   Speaker
 HadThing:  Car
```

Lógica Descriptiva

Un subconjunto de Lógica del primer orden, especializada en generar conceptos

- Lógica de primer orden

  HombreFeliz(x) ⇔ TieneSalud(x) ^ TieneDinero(x) ^ TieneAmor(x)

- Lógica descriptiva

  HombreFeliz = Y(TieneSalud, TieneDinero, TieneAmor))

Los más común: **ALC** (***Attributive Language with Complements)***

- **TBox**: Define tipos y Relaciones
- **ABox**: Define individuos

TBOX	**ABOX**
Mujer ≡ Persona _ SexoFemenino	MadreSinHija(Elena)
Hombre ≡ Persona _ ¬Mujer	Father(Pedro)
Madre ≡ Mujer _ ∃tieneHijo.Persona	hasChild(Elena, Pedro)
Padre ≡ Hombre _ ∃tieneHijo.Persona	hasChild(Pedro,Lucas)

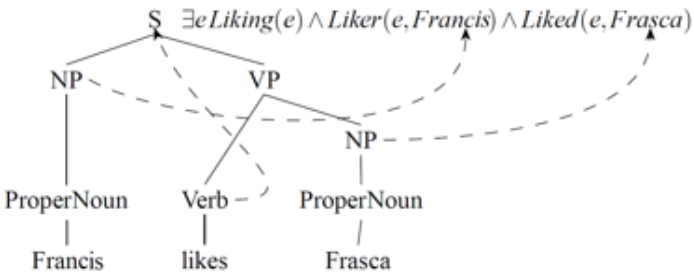
7. Análisis Semántico

Hay diferentes formas de análisis semántico. Todas basen en **"el principio de composición":** el significado de una oración se construye a partir del significado de sus partes.

Análisis dirigido por la sintaxis

El análisis se hace paso a paso, ampliando el conocimiento previo.

Al Árbol sintáctico se añadirán **anotaciones semánticas** y se aplican en función Del Árbol sintáctico.

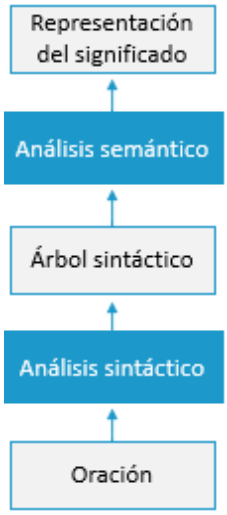


Reglas léxicas

a) NP → Matías
b) N → restaurante
c) Det → un
d) Vt → abrió

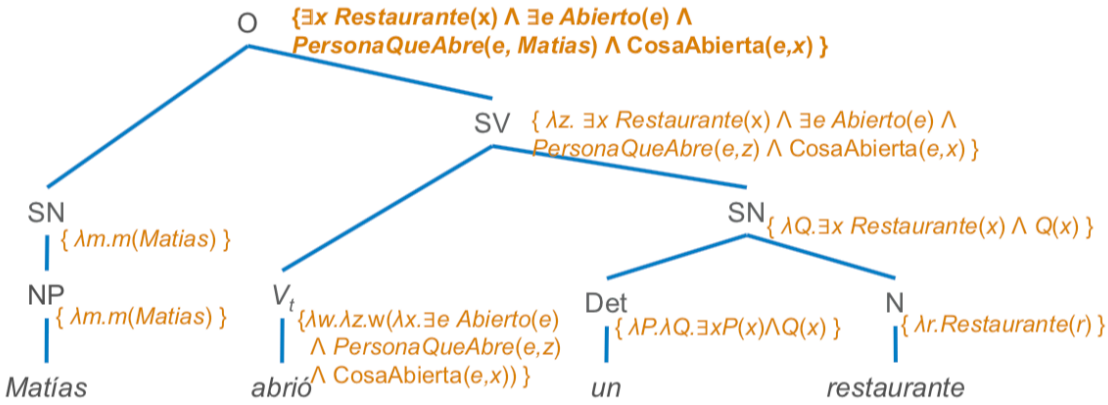
Reglas gramaticales

1) O → SN
2) SN →NP
3) SN → Det N
4) SV → Vt SN



Análisis semántico integrado con el análisis sintáctico

Un análisis sintáctico puede producir múltiples árboles sintácticos (por ambigüedad). Esta ambigüedad se puede resolver, haciendo un análisis sintáctico y semántico al mismo tiempo. ¡Se crear el árbol paso a paso siguiendo ambos tipos de reglas!



Anotación semántica (sirve para ambos métodos)

1. Asociar expresiones del cálculo lambda complejas, similares a una función, a las reglas léxicas. [anota las reglas léxicas con expresiones del cálculo lambda]
2. Copiar el valor semántico del único constituyente a la construcción de la regla gramatical solo tiene un constituyente. [Sustituye el contenido gramático en las reglas gramaticales]
3. Aplicar la semántica de uno de los constituyentes de una regla gramatical a las semánticas de otro de los constituyentes de la regla como si fuera una función. [juntando reglas gramáticas, utiliza el contenido gramático de uno como parámetro de la función del otro]

8. Semántica Léxica

Es el estudio del significado y las relaciones de sentido entre palabras.

Lema: Es la versión básica de una palabra

Relaciones entre lemas

Homonimía: palabras similares con sentidos distintos [banco[1], banco[2]]
Homografía: escrito igual, sentido distinto [banco[1], banco[2]]
Homofonía: escrito distinto, sueña igual, sentido distinto [vello, bello]
Sinónima: palabras con lemas similares [comenzar, empezar]
Antónimo: palabras con lemas opuestas [grande, pequeño]
hipónimo: una palabra es un subconjunto del otro [perro, animal]
hiperónimo: revés de hipónimo [animal-> perro]
merónimo: sea parte de algo [rueda -> coche]
metonimia: algo sobre otra palabra que tiene un sentido relacionado
polisémicos: dos sentidos de una palabra están relacionado

Resolver la ambigüedad en el análisis semántico

Aprendizaje supervisado

Basado en datos etiquetadas entrena un modelo por palabra o múltiples, para que extrae características que sirven para predecir el sentido de la palabra.
Características pueden ser

- **Colocación**: el contexto de la palabra (las palabras alrededor; en orden)
- **Palabras vecinas**: Bag-of-words; un conjunto de palabras, también el contexto más probable de la palabra.

Supervisado débil (o basado en conocimiento)

Utilizan bases de conocimiento (diccionario o tesauros), y las firmas (definiciones) de las palabras en estos bases para elegir un sentido entre los posibles.

Lesk Simplificado: Comparar el contexto con la firma y contar las palabras similares
Lesk: Comparar la firma de la palabra con la firma de cada palabra del contexto
Corpus Lesk: Comparar la firma con el texto completo en cual aparece la palabra ambigua. Utilizar un peso (IDF) en vez de # de overlap.

Semisupervisado

Se necesita un conjunto de datos etiquetado pequeño que se amplia durante el entrenamiento.
El algoritmo de **Yarowsky** se utiliza recursivamente, prediciendo un el significado de palabras no etiquetados basado en los datos entrenado. Las palabras etiquetadas más probables (más correcto) se añaden a los datos de entrenamiento.
Hay dos heurísticas para definir el sentido de las palabras etiquetados iniciales

- **Por colocación**: tener múltiples firmas; una por sentido
- **Por discurso**: tener una única firma por texto; elegir un único sentido por texto

No supervisado

Algoritmo de Schütze
Calcula en contexto para cada instancia de una palabra en un corpus, y hace un clustering de estos vectores de contexto. Los centroides de estos clusters sirven como referencia de sentido.

Para una palabra ambigua, elija el sentido que es el centroide más cerca al vector del contexto de la palabra.

Como definir la similitud entre palabras?

- Métodos estadísticos basados en la probabilidad de coocurrencia en un corpus
- Basado en relaciones en un tesauro

En un Tesauro

Distancia del camino

Creando una red de sentidos basado en la información del Tesauro, se calcula la similitud de dos sentidos como distancia en esta red.

- La similitud en función de la distancia entre los significados

$$sim_{path(c_1,c_2)} = \frac{1}{pathlen(c_1, c_2)}$$

- La de dos palabras como la máxima distancia de sus significados

$$wordsim(w_1, w_2) = \max_{\substack{c_1 \in senses(w_1) \\ c_2 \in senses(w_2)}} sim(c_1, c_2)$$

Información compartida

Para dos significados, calcula el ancestro común más cercano en la red de sentido del tesauro. (**LCS, *lowest common subsumer)***

Calcula la información que lleva un sentido basado en la frequencia de occurencia (**P**).

$$P(c) = \frac{\sum_{w \in words(c)} count(w)}{N}$$

$$sim_{resnik(c_1,c_2)} = -\log P(LCS(c_1, c_2))$$

words(c) … todos los palabras en el Tesauro debajo de c (hijos/descendientes).

9. RECURSOS PARA RLN

Tipos de Diccionarios

Diccionario: palabras con definición
Lexicón: diccionario + información morfológica (elementos de una palabra)
Tesauro: diccionario + algunas relaciones entre palabras (estructura jerárquica) (sinónimos, hiperonimia/hiponomia, meronima/holnima)
Base de datos: Tesauro con todos los tipos de relaciones entre palabras

Example: WordNet

Una base de datos de lenguaje Ingles más compleja (1998).
Estructurado en **synsets** (palabras distintas)

Corpus lingüístico

Colección de textos, públicos online desde los 1960s.

- Brown Corpus (Ingles)
- CREA (español, amplia variación de textos y transcriptos)
- CORPES (Español/Latino)
- IMPACT-es: 86 obras clásicas de Cervantes

Liberías

NLTK (Python)

Servicios Online

Google Cloud Natural Language
Amazon Comprehend
IBM Watson NLU
Microsoft Language Understanding (LUIS)

10. Agentes Conversacionales

Hay dos tipos de agentes conversacionales

- **Dedicados:** dedicados a contextos/temas específicos
- **Chatbots:** permiten conversaciones no estructuradas y de cualquier contexto

Características de las conversaciones entre humanos

- **Acto de habla**: hay diferentes tipos de habla (actos de cambian el estado del mundo, sugerencias, ordenes, preguntas
- **Turno de palabra**: el interlocutor de cambian en 'transition-relevance places'
- **Pares adyacentes**: hay estructuras en cada conversación (Preguntas-respuestas)
- **Puntos de coincidencia**: feedback para el locutor que sea entendido
- **Informaciones implícitas**:

Estructura de los agentes conversacionales

1. **Reconocimiento del Habla**:
2. **Compresión de LN**: extraer semántica, contexto, sentido, intención, puntos de coincidencia
3. **Gestión de diálogo**
    - Agentes basados en frames:
    - Agentes basados en diálogo
4. **Generación de lenguaje natural:**
    - Que decir
    - Como decirlo
5. **Conversión del texto al habla:** concatenación de trozos grabados

Agente conversacional basado en frames

La mayoría de los agentes actuales utilizan frames + gramática semántica para el análisis del habla. El estado interno del agente es una **máquina de estados finitos**
Para cada hueco del frame, el agente tiene una frase/pregunta.

+ Siempre saben donde están en la conversación
+ Siempre tienen una pregunta/frase que decir
- No están muy flexible o genéricos

Agente conversacional basado en el diálogo

Aplican estrategias más complejas (que los de los frames) como contextos locales, decisión de Markov, aprendizaje por refuerzo.

+ Sacan información de cada interacción con el usuario, no solo que necesitas para rellenar el siguiente hueco
+ Pueden predecir/modular que acción es la más probable para cada turno de palabra
- No dirigen la conversación hacia un gol

Tipos de Chatbots

Chatbots basados en reglas

Son los primeros tipos de chatbots; que no tienen ningún entendimiento semántico.

- **reglas patrón -> transformación (**Ejemplo: ELIZA)

Chatbots basados en corpus

Basado en un corpus de conversaciones humanos

- **Basados en recuperación de información**: Contestan en función de conversaciones similares en su corpus.
- **Secuencia a Secuencia**: Paradigmas de traducción automática