

# ***NLP***

## Preprocesamiento de texto

Dr. Rodrigo Cardenas Szigety  
rodrigo.cardenas.sz@gmail.com

# Programa de la materia



**Clase 1:** Introducción a NLP, Vectorización de documentos.

**Clase 2:** Preprocesamiento de texto, librerías de NLP y Rule-Based Bots.

**Clase 3:** Word Embeddings, CBOW y SkipGRAM, representación de oraciones.

**Clase 4:** Redes recurrentes (RNN), problemas de secuencia y estimación de próxima palabra.

**Clase 5:** Redes LSTM, análisis de sentimientos.

**Clase 6:** Modelos Seq2Seq, traductores y bots conversacionales.

**Clase 7:** Celdas con Attention. Transformers, BERT & ELMo, fine tuning.

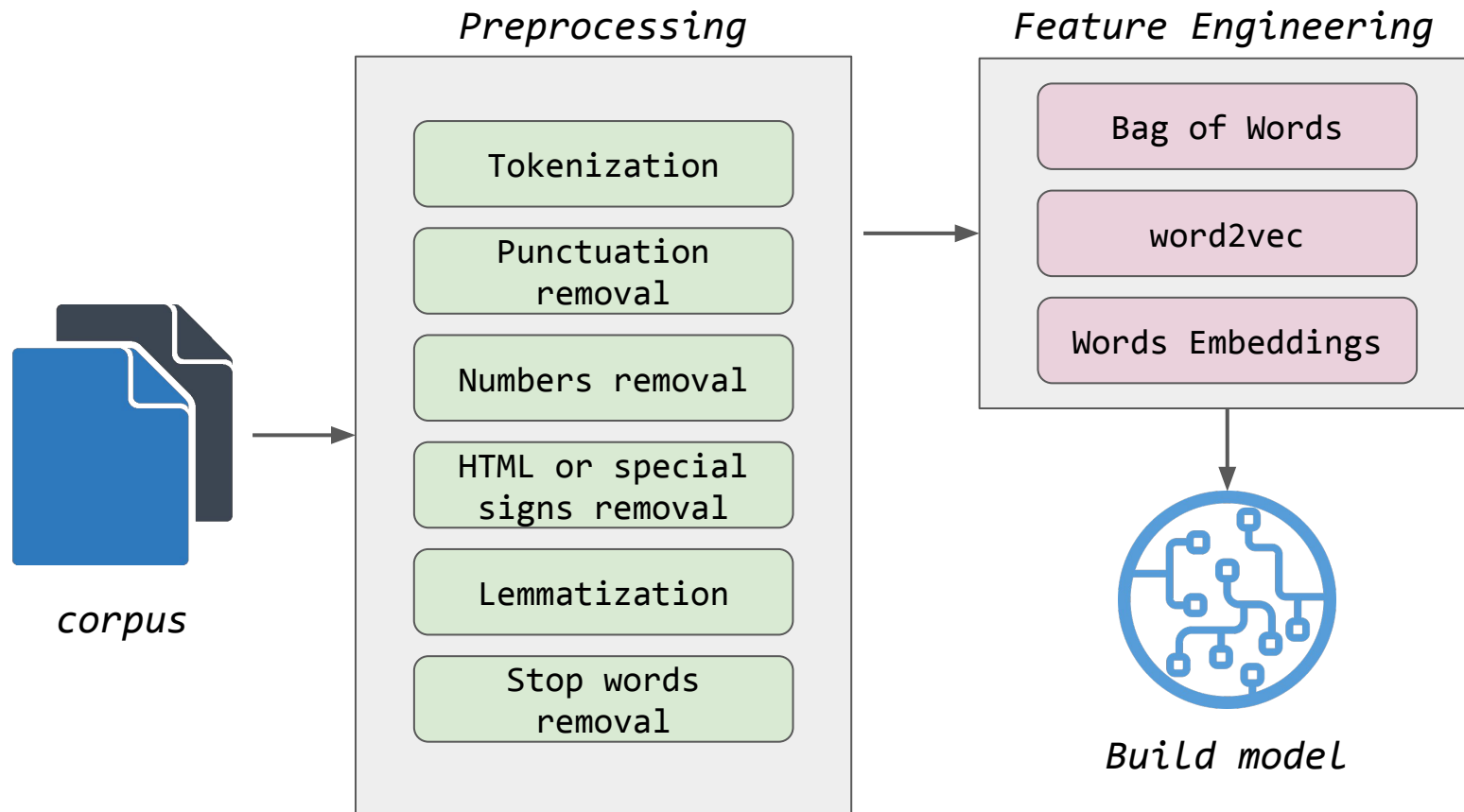
**Clase 8:** Cierre del curso, NLP hoy y futuro, deploy.

\*Unidades con desafíos a presentar al finalizar el curso.

\*Último desafío y cierre del contenido práctico del curso.

# Preprocesamiento de texto

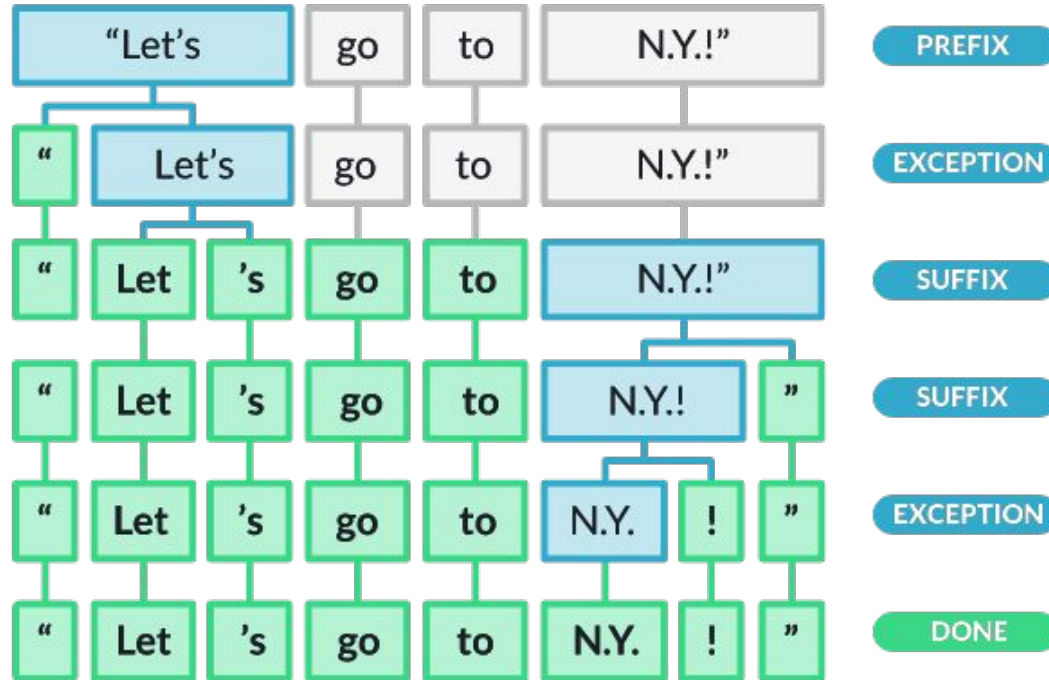
[LINK GLOSARIO](#)



# Tokenizar



*"Proceso en el cual una sentencia o documento es dividida en palabras o términos individuales. Los símbolos son dejados aparte para luego tratarlos"*

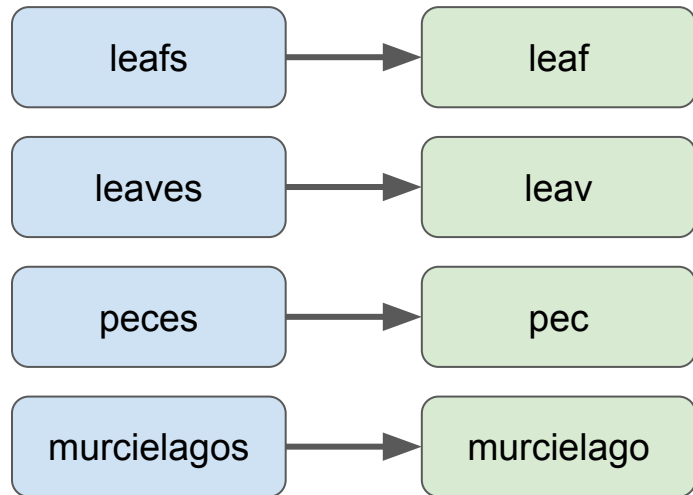


# Derivado (steeming)



*"Devolverá el tallo de una palabra, que no necesita ser idéntica a la raíz morfológica de la palabra"*

*Elimina los sufijos*



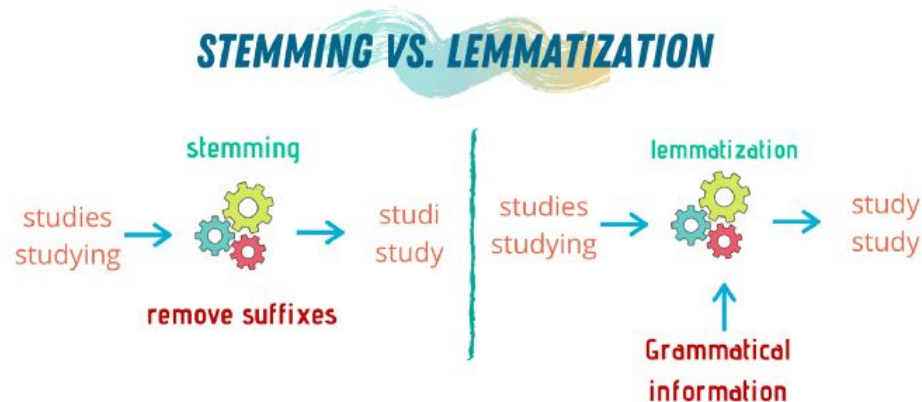
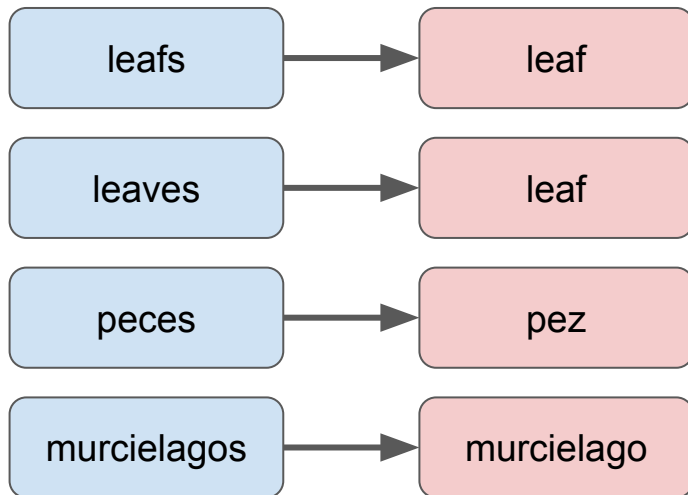
*plural irregular*

*especimen, especímenes  
régimen, regímenes*

# Lematización (lemmatization)



*"Devolverá la forma diccionario de una palabra, su raíz"*



# Parts-of-speech (POS) - Tagging



"POS es el proceso de identificar cada término en la oración, tageandolos como **sustantivo (noun)**, **verbo (verb)**, **adjetivo (adj)**, etc"



# Stop words

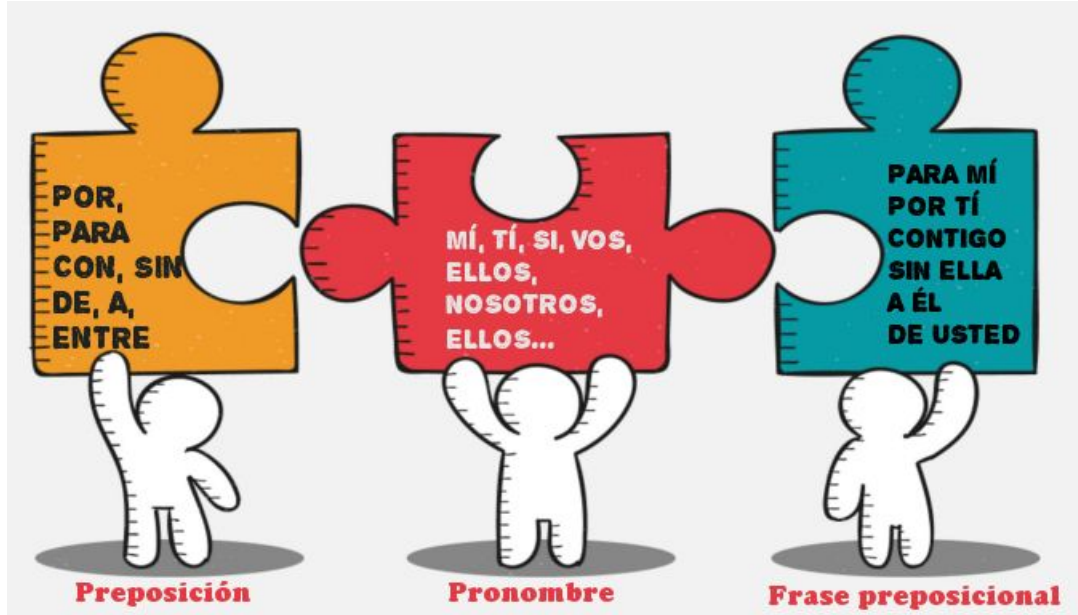


*"Palabras que no aportan valor al significado de una oración ya que son muy frecuentes o comunes en el lenguaje"*

artículos

pronombres

preposición



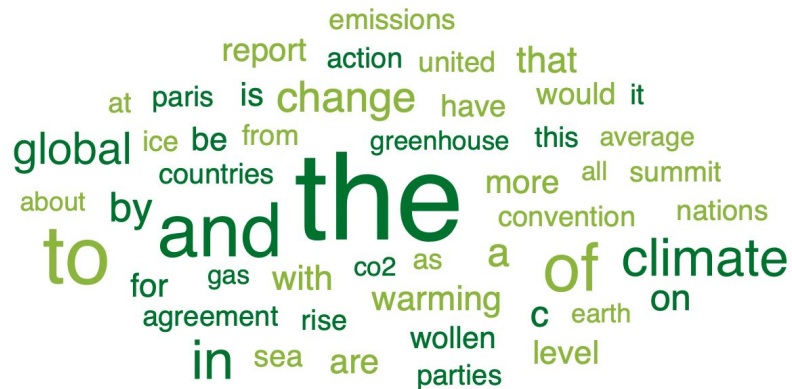


# Stop words

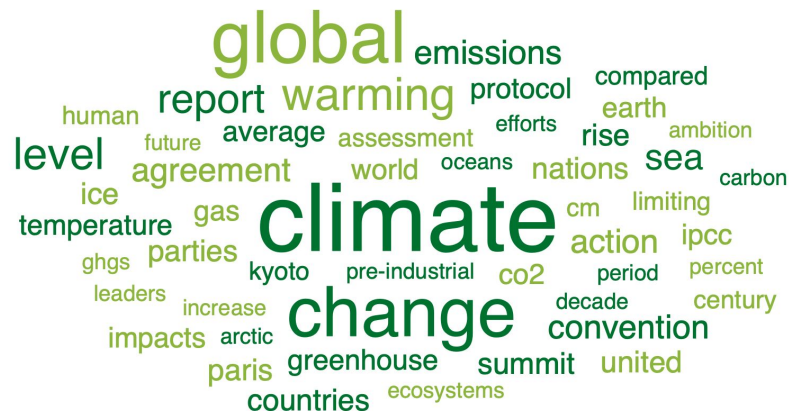


*Analizar un texto relacionado con calentamiento global (global climate)*

*Texto con Stop Words*



*Texto sin Stop Words*



# Librerías de NLP

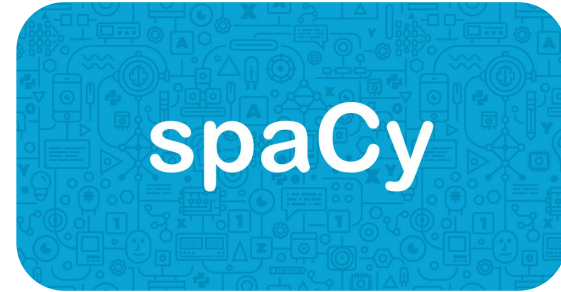


Gran comunidad de desarrollo

No soporta GPU

Más optimizada

Más lenta en gran volúmenes de datos o operaciones



Más moderna e implementa los últimos features

Soporta GPU

Menos optimizada

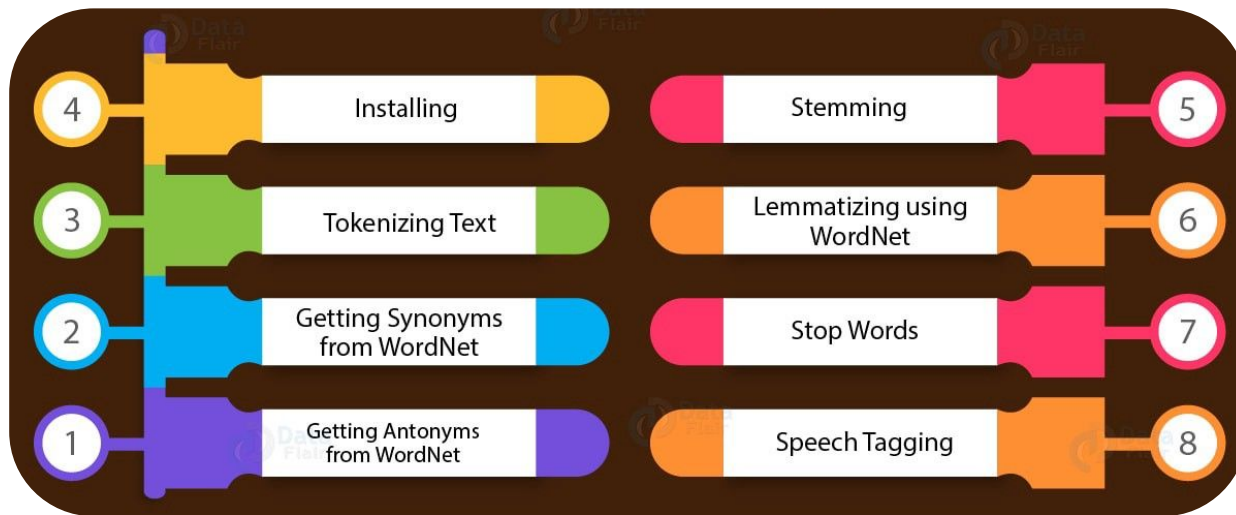
Más rápida en gran volúmenes de datos o operaciones

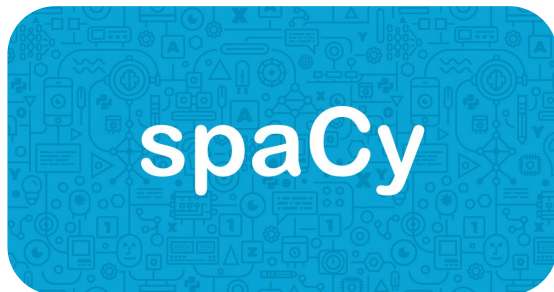


*"Librería por excelencia de procesamiento de lenguaje natural para Python"*

*Inicios 2009*

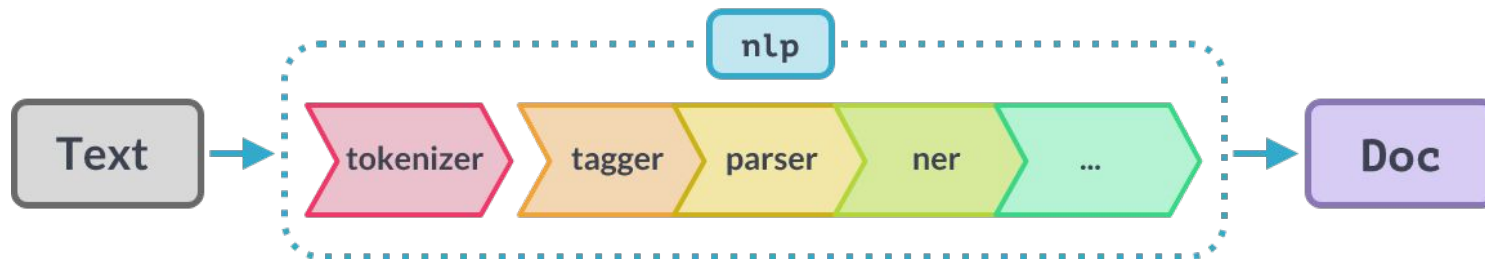
*Implementa una tool/algoritmo para cada etapa de preprocesamiento de NLP*





*Inicios 2015*

- ✓ Support for **64+ languages**
- ✓ **63 trained pipelines** for 19 languages
- ✓ Multi-task learning with pretrained **transformers** like BERT
- ✓ Pretrained **word vectors**
- ✓ State-of-the-art speed
- ✓ Support for custom models in **PyTorch**, **TensorFlow** and other frameworks

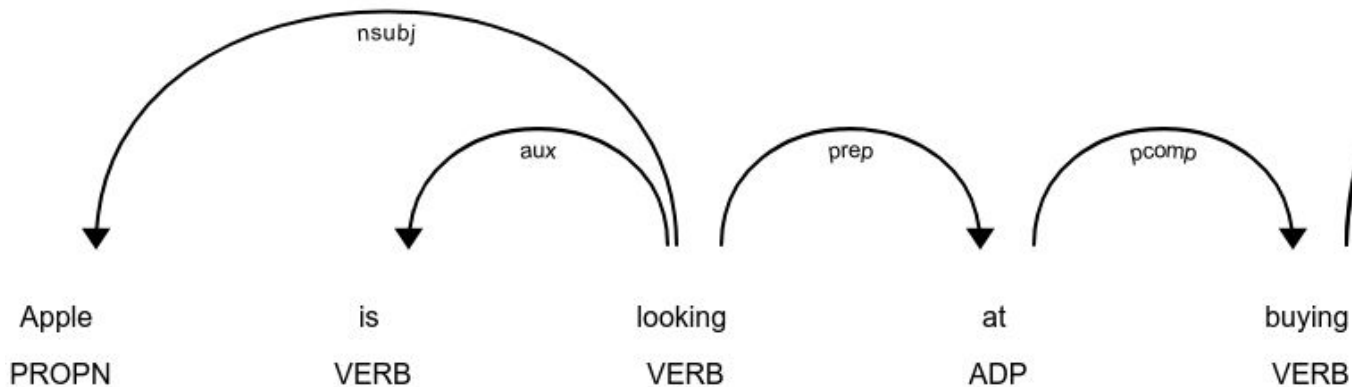




```
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")

for token in doc:
    print(token.text, token.lemma_, token.pos_, token.tag_, token.dep_,
          token.shape_, token.is_alpha, token.is_stop)
```



# Un resumen de todo lo visto

POS

TAG

DEP



TEXT	LEMMA	POS	TAG	DEP	SHAPE	ALPHA	STOP
Apple	apple	PROPN	NNP	nsubj	Xxxxx	True	False
is	be	AUX	VBZ	aux	xx	True	True
looking	look	VERB	VBG	ROOT	xxxx	True	False
at	at	ADP	IN	prep	xx	True	True
buying	buy	VERB	VBG	pcomp	xxxx	True	False
U.K.	u.k.	PROPN	NNP	compound	X.X.	False	False
startup	startup	NOUN	NN	dobj	xxxx	True	False
for	for	ADP	IN	prep	xxx	True	True
\$	\$	SYM	\$	quantmod	\$	False	False
1	1	NUM	CD	compound	d	False	False
billion	billion	NUM	CD	pobj	xxxx	True	False

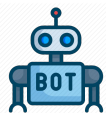


Link al Colab

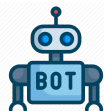


*LINK*

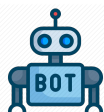
# BOTs lingüístico



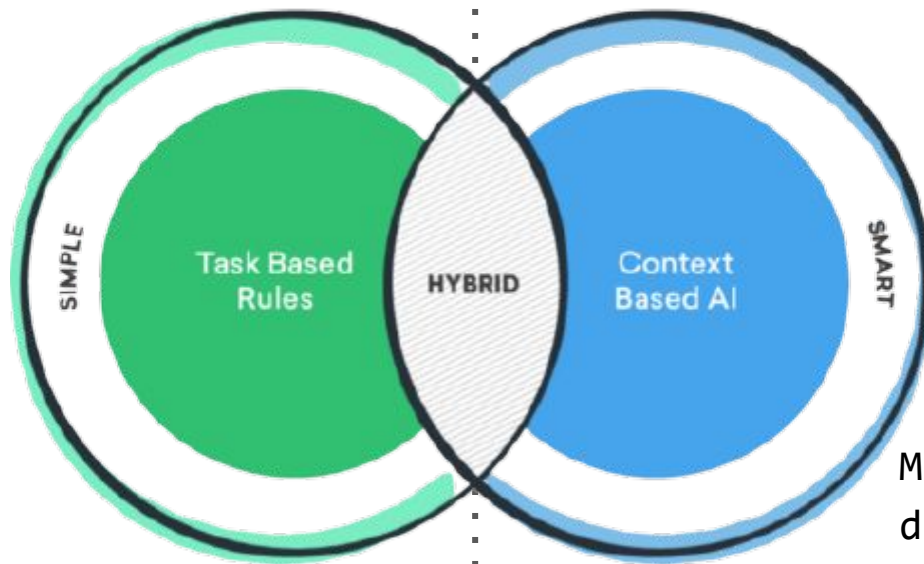
Son limitados  
a una tarea  
específica



Fácil y  
"barato" de  
entrenar



Ideal para  
chats de  
pedidos



Interactúan  
casi como un  
humano



Más difíciles  
de entrenar,  
requieren más  
datos y cómputo

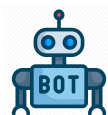
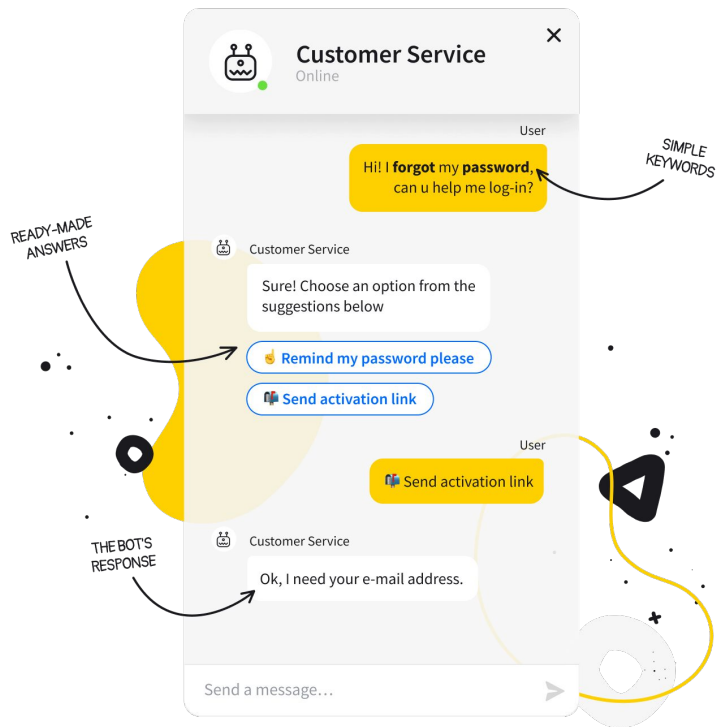


Ideal para  
asistentes  
virtuales

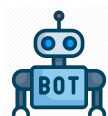




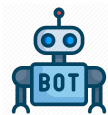
# Rule-Based BOT



Nuestro bot será entrenado con <TAGS> (ej: saludo)



Cada <TAG> será representado por un patrón de posibles preguntas <patterns> (X)



Cada <TAG> tendrá uno o varias posibles respuestas <classes> (y)



Link al Colab



*LINK*



Link al Colab



*LINK*



Tomar un ejemplo de  
los bots utilizados

Construir el propio





# ¡Muchas gracias!