



DataSpartan



Construyendo un chatbot basado en Inteligencia Artificial

En este workshop daremos nociones de la lógica detrás de cada uno de los componentes principales de un chatbot y proporcionaremos guía para construir uno.



[https://github.com / rorisDS / workshop_chatbot](https://github.com/rorisDS/workshop_chatbot)



DataSpartan

Expertos en las áreas de Inteligencia Artificial (IA), Big Data, Investigación Cuantitativa, Computación de Alto Rendimiento (HPC), Machine Learning (ML) y Modelización de Riesgos.



PYMES

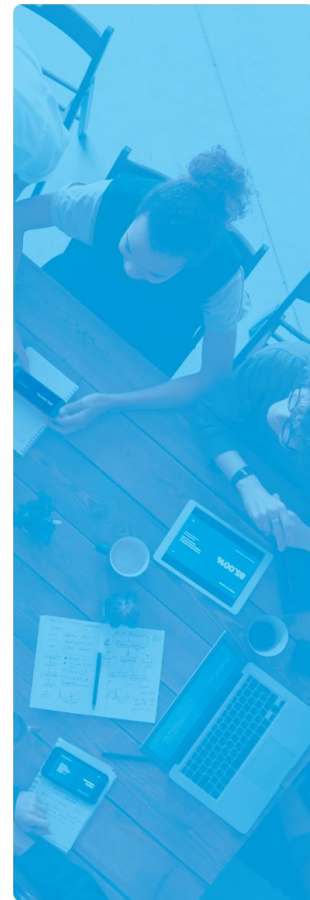
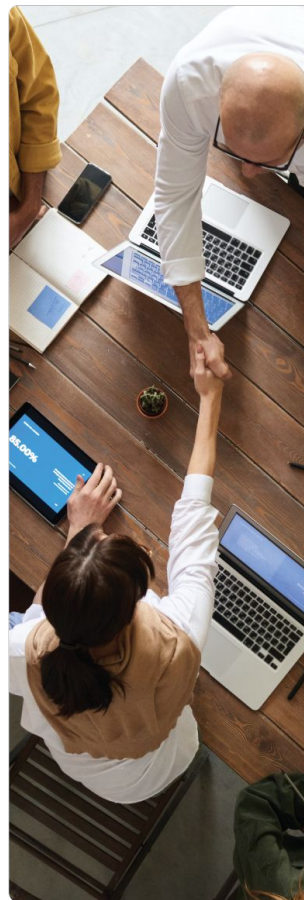


EMPRENDEDORES



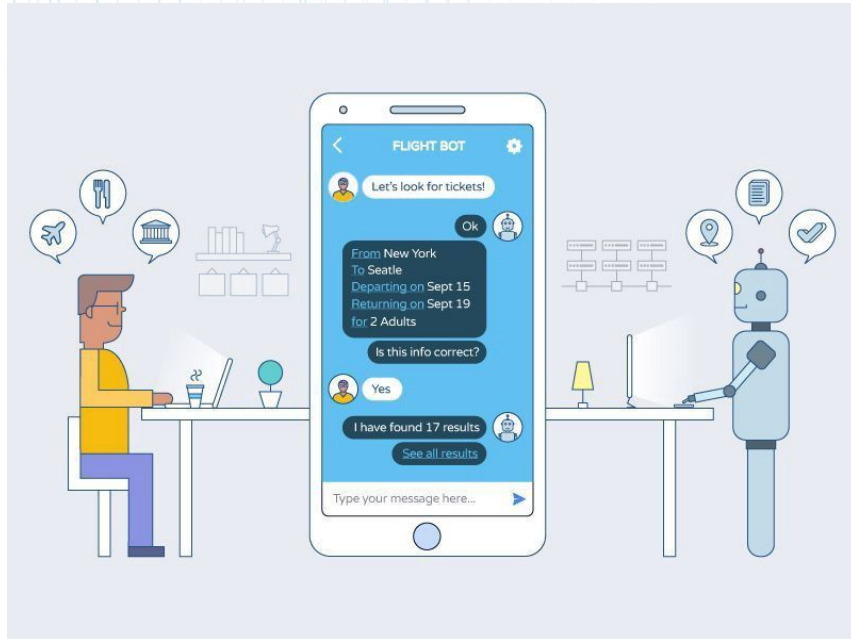
EQUIPOS I+D

Investigamos, diseñamos, construimos, implementamos y validamos soluciones, para facilitar que las empresas puedan alcanzar su potencial, de forma más rápida y asequible de lo que sería posible contando únicamente con sus propios medios.



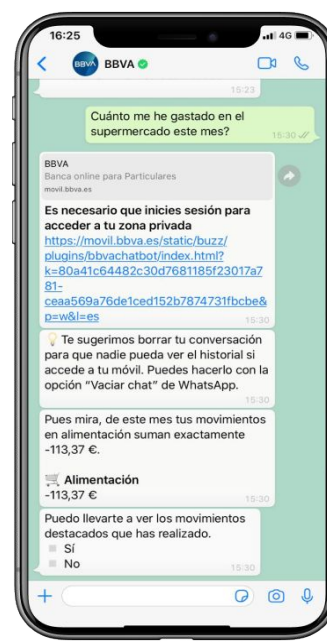
Chatbot o asistente conversacional

Un chatbot es un agente software que permite automatizar conversaciones en **lenguaje natural** (y/o la ejecución de acciones en base a mensajes).



- Asistentes: Siri, Alexa, Cortana, ...
- Atención al cliente / FAQs
- Gestionar área de clientes (consumos, envíos, compras, subastas, ...)
- Tutor educativo / culinario / salud ...
- Consultar previsión tiempo / bolsa / ...
- Gestor domótico
- ...

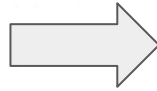
Ejemplos de chatbot



Lenguaje Natural

Lenguaje natural es la lengua o idioma hablado o escrito por humanos para propósitos generales de comunicación (ex. español, francés o inglés).

- Ambiguo
- Desestructurado



difícil de procesar por máquinas

NLP (Natural Language Processing)

Inteligencia Artificial \Rightarrow comprensión y manipulación del lenguaje humano

Natural Language Processing (NLP)

Larga historia ⇒ 1950 - Hoy en día

Avance histórico por aplicación de Redes Neuronales ⇒ Transformers

Aplicaciones:

- Categorizar textos (ej. Sentiment Analysis) ([demo](#))
- Question Answering ([demo](#))
- Summarization ([demo](#))
- Generación de texto ([demo](#))
- Similitud de frases ([demo](#))

Recursos:

- Tutorials: [link](#), [link](#), [link](#)
- Frameworks: [spaCy](#), [NLTK](#), [AllenNLP](#)
- Machine Learning library: [HuggingFace](#)

The screenshot shows a web interface for a Question Answering demo. At the top, it says "Question Answering demo" with a lightning bolt icon, followed by "using deepset/roberta-base-squad2". Below this is a dropdown menu labeled "Question Answering" with "Examples" selected. A text input field contains the question "Where do I live?". To the right of the input is a "Compute" button. Below the input is a "Context" section with a text area containing "My name is Sarah and I live in London". Under the context, it says "Computation time on cpu: cached". The answer "London" is displayed in a green box, with a score of "0.777" to its right. At the bottom, there is a "JSON Output" button and a "Maximize" button.

Question Answering demo
using `deepset/roberta-base-squad2`

Question Answering Examples

Where do I live? Compute

Context

My name is Sarah and I live in London

Computation time on cpu: cached

London 0.777

JSON Output Maximize

Natural Language Understanding (NLU)

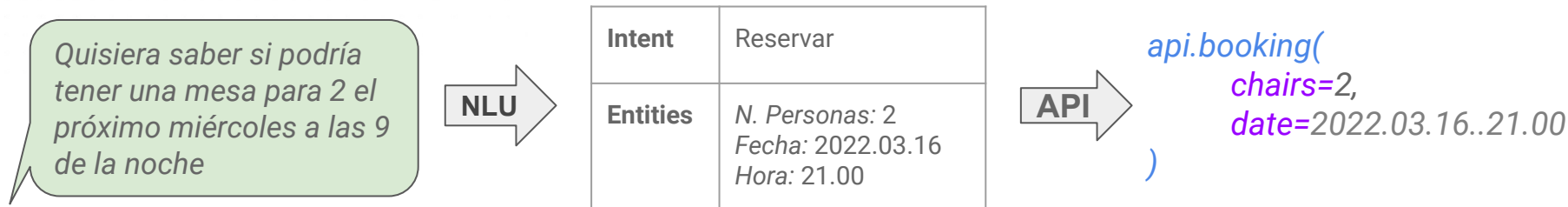
Interpretar los mensajes del usuario:

- **Intent / Propósito:**

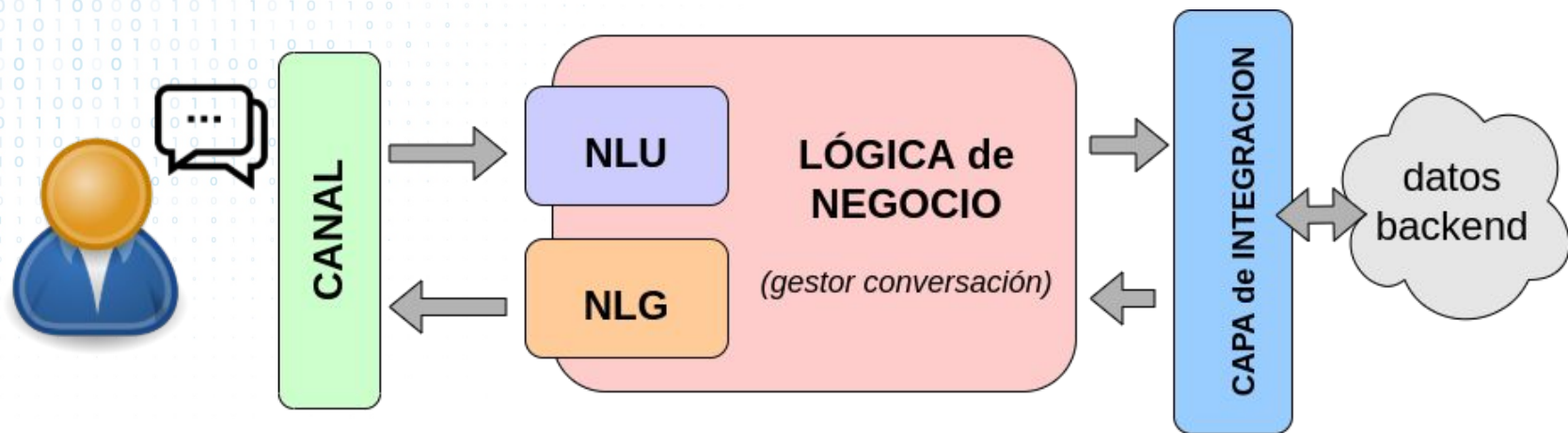
Significado general de un mensaje. ¿Cuál es la intención del usuario?

- **Entities / Entidades:**

Información adicional o modificadores del mensaje



Arquitectura general



Implementamos un chatbot básico (I)

Ejemplo básico \Rightarrow Objetivo didáctico

Chatbot profesional \Rightarrow Usa librerías: [Rasa](#), [Wit AI](#), [Dialogflow](#), entre [otros](#).

No reinventar la rueda



Nuestro chatbot



Chatbot hecho
con librería

Implementamos un chatbot básico (II)

Bea - Asistente cafetería de teleco

- Consultar horario
- Consultar menú
- Hacer una comanda



Implementamos Bea

Mismo core: Backend, NLG y Lógica de Negocio

3 versiones de NLU:

- Basado en patrones
- Basado en similitud de oraciones
- Basado en Rasa NLU

Código Python: [Google colab](#)

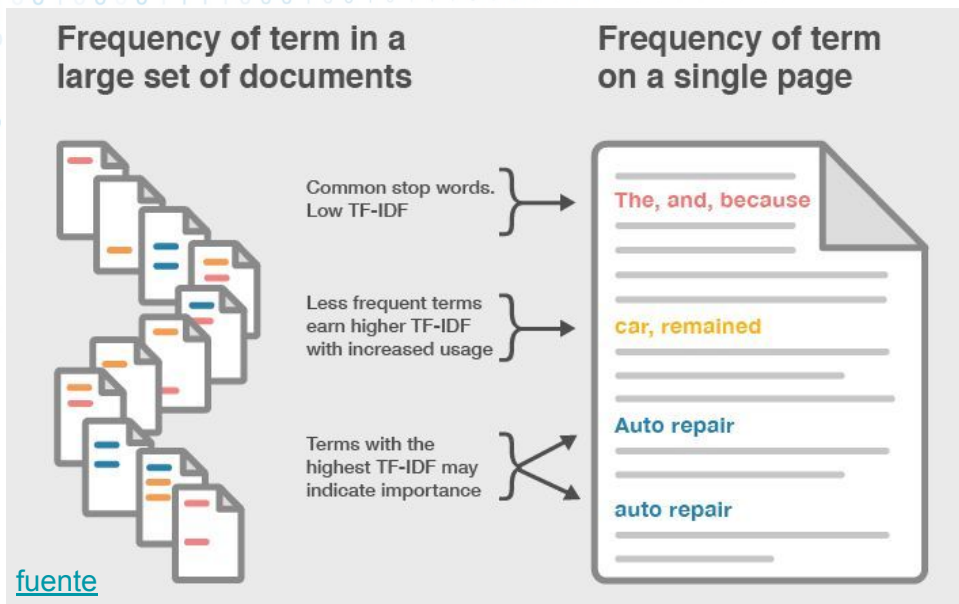
[Por qué usamos Python en ML?](#)

[Qué es jupyter notebook?](#)

Bea basado en patrones (I)

Términos identificativos de cada *Intern*

⇒ *Principios de TF-IDF*



$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

$tf_{i,j}$ = number of occurrences of i in j
 df_i = number of documents containing i
 N = total number of documents

Bea basado en patrones (II)

- Muy simple
- Proclive a errores
 - No identificar el *Intent* de los mensajes
 - ¿Cuál es la agenda de la cafetería? ⇒ sin término “horario”
 - Uso del mismo patrón en otro *Intent*
 - ¿Quisiera un plato de vuestro menú? ⇒ término “menú” para pedir

Bea basado en similitud de oraciones (I)

Texto a representación del significado

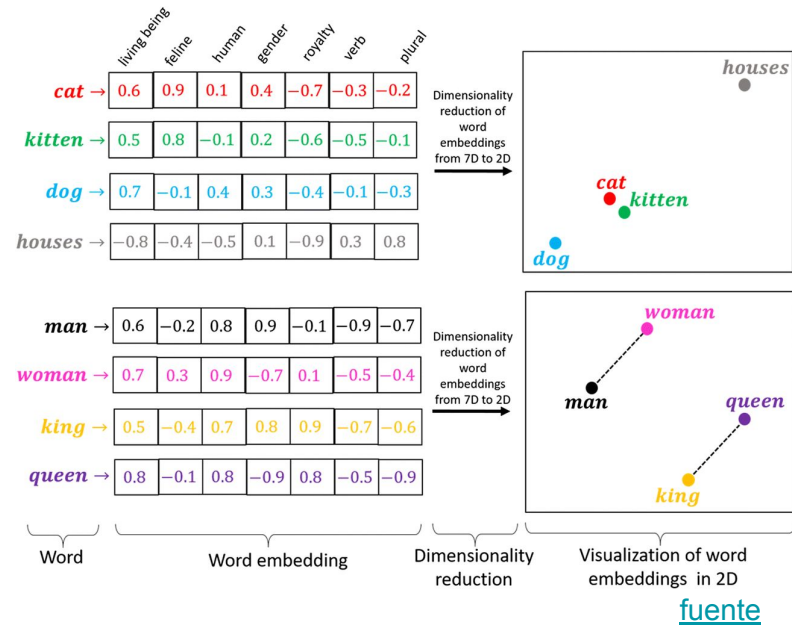
⇒ text embeddings

⇒ embedding es un vector

⇒ un vector es un punto en el espacio

Similitud entre vectores (cosine):

$$\text{cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$



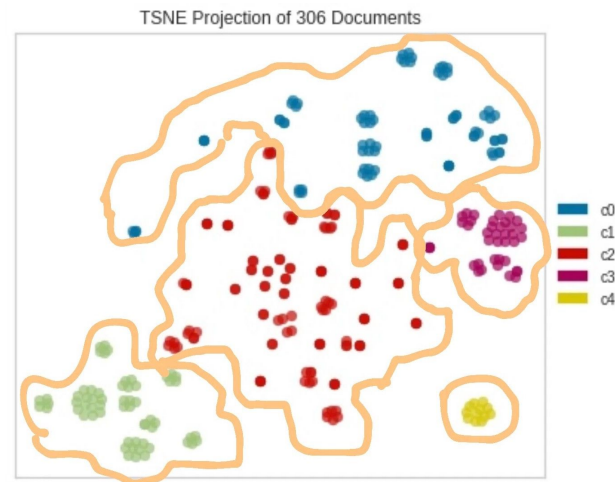
Bea basado en similitud de oraciones (II)

Listar colección de ejemplos por *Intent*

```
- intent: bye
examples: |
  - adios
  - bye
  - chao
```

Principios de *clustering*:

- Embeddings agrupados por *Intent*
- El mensaje caería dentro del área de un *Intent*



Bea basado en similitud de oraciones (III)

- Mejora la identificación de *Intents*
- Deficiencias:
 - Identificación de *Entities*
 - Necesidad de gran variedad de ejemplos
 - Se parte de embeddings/modelos pre-entrenados
 - No se refina (o *fine-tuning*) en base al dominio

Otro ejemplo de chatbot basado en similitud: [link](#)

Bea basado en Rasa NLU (I)

Componente de [RASA](#)

⇒ identificación de *Intents* y *Entities*

Datos de entrenamiento:

- Fichero [YAML](#)
- [Formato específico](#)
- [Datos de entrenamiento extra:](#)
 - Sinónimos
 - Tablas de lookup
 - Expresiones regulares

```
nlu:  
- intent: greet  
  examples: |  
    - Hey  
    - Hi  
    - hey there [Sara](name)  
  
- intent: faq/language  
  examples: |  
    - What language do you speak?  
    - Do you only handle english?
```


Bea basado en Rasa NLU (II)

Configuración basada en pipelines

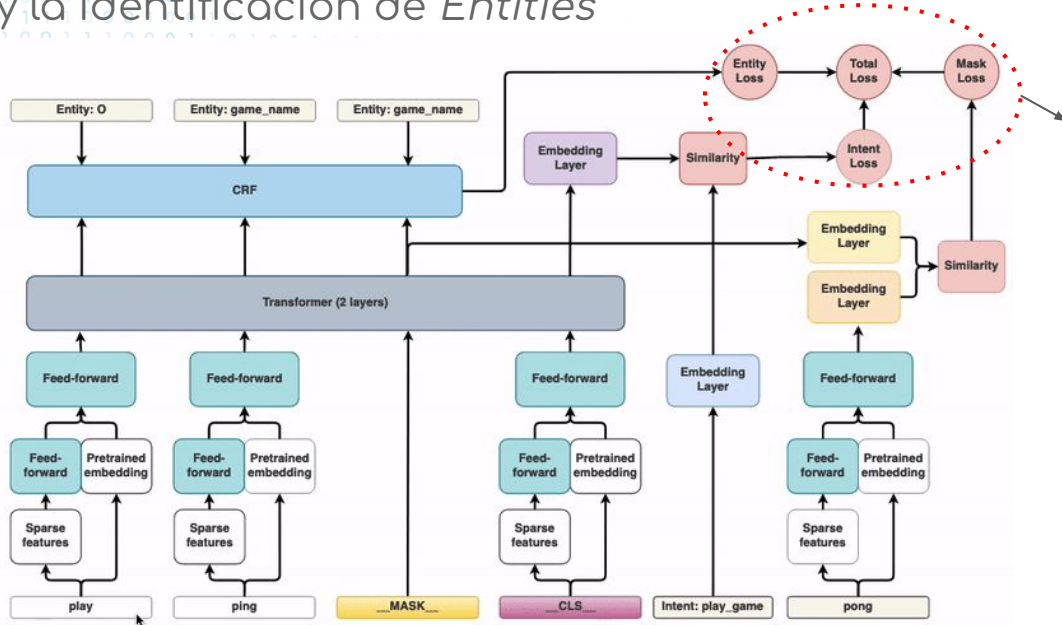
- ⇒ secuencia de componentes
- ⇒ configuración de cada componente
- ⇒ componentes para diferentes procesos
 - Language Models
 - Tokenizers
 - Featurizers
 - Intent Classifiers
- ⇒ puedes crear tus propios componentes

```
# Configuration for Rasa NLU.  
# https://rasa.com/docs/rasa/nlu/components/  
language: sp  
pipeline:  
- name: WhitespaceTokenizer  
- name: RegexFeaturizer  
- name: LexicalSyntacticFeaturizer  
- name: CountVectorsFeaturizer  
- name: CountVectorsFeaturizer  
  analyzer: "char_wb"  
  min_ngram: 1  
  max_ngram: 4  
- name: DIETClassifier  
  epochs: 100  
- name: EntitySynonymMapper  
- name: ResponseSelector  
  epochs: 100
```

Bea basado en Rasa NLU (III)

DIET (Dual Intent and Entity Transformer)

Arquitectura *Transformers* que permite manejar tanto la clasificación de *Intents* y la identificación de *Entities*



Entrenamiento basado en el cálculo de *Intent*, *Entities* y *Mask* loss al mismo tiempo

[Video explicativo](#)

[Más info y ejemplos](#)

Bea basado en Rasa NLU (IV)

- Identificación de *Intents* y *Entities* usando AI
 - Se entrena/refina un modelo para el dominio de aplicación
 - *DIET Classifier* es State of The Art (*principios del 2022*)
- Problemática:
 - Necesidad de disponer de ejemplos exhaustivos
 - Configuración compleja:
 - Variedad de componentes
 - Definir la secuencia de componentes
 - Configuración por componente



PROBLEMÁTICA
MACHINE LEARNING

Otro ejemplo de chatbot basado en RASA NLU: [link](#)

Conclusiones

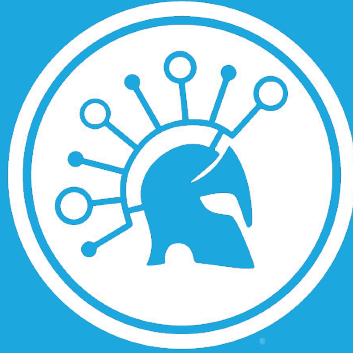
Chatbot es un proceso lógico

basado en interacción de componentes

desde más simples a más complejos \Rightarrow resultados en consonancia

Bea es un chatbot simple

\Rightarrow permite entender chatbots complejos (RASA o Alexa)



DataSpartan

Réтанos hoy
informacion@dataspartan.com

www.dataspartan.es