

Jerónimo Molina Molina

Procesamiento del Lenguaje Natural

7. Hugging Face

Es cierto que siempre se van a surgir necesidades específicas que, posiblemente, nos conduzcan a entrenar modelos concretos, pero es también una gran verdad que hay una cantidad ingente de modelos, ya entrenados, que pueden utilizarse de forma directa (o en su defecto tras un proceso de Transfer Learning o de Fine Tuning para que se ajusten mucho mejor a nuestro objetivo).

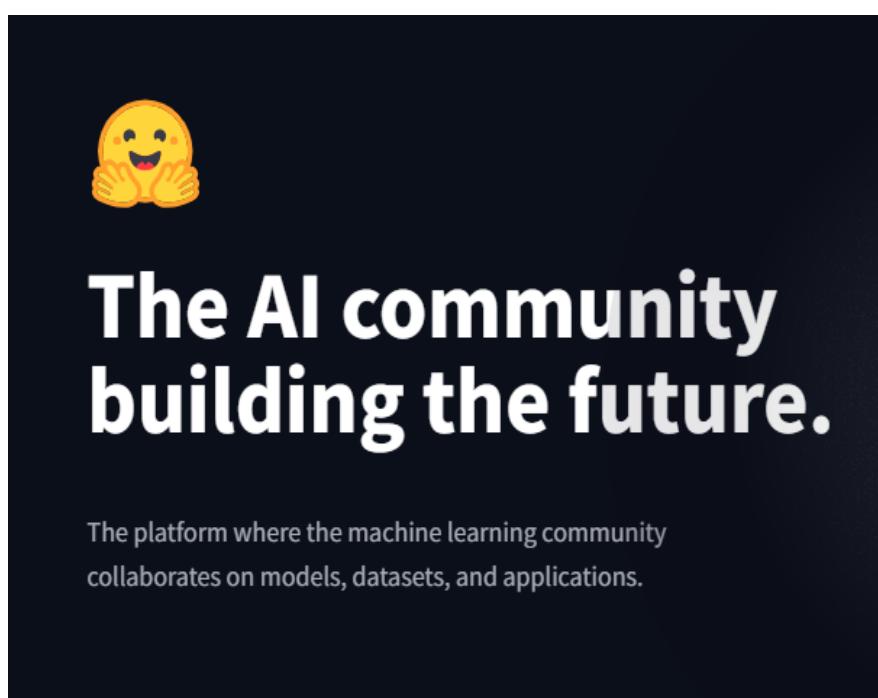
Y es que no todo son grandes plataformas con macro servicios que esperan atender todas las posibles necesidades, sino que existen plataformas de las que descargar modelos concretos, para tareas específicas, que pueden suponer un cambio drástico en la funcionalidad de los proyectos que abordéis.

Hoy quiero presentaros **Hugging Face**, una plataforma online en la que no solo encontraréis numerosos modelos pre entrenados, sino que también podréis contratar un servicio para subir los vuestros propios y emplearlos como si de una REST API se tratase.

Hugging Face

Es una empresa americana, fundada por dos franceses (Julien Chaumond y Clément Delangue) y un americano, Thomas Wolf, fundada en 2016 y que tras varios intentos de negocio se centró en convertirse en una plataforma online dedicada al machine learning y centrada en el procesamiento del lenguaje natural (inicialmente) para luego abrirse a otros campos de la IA, como el de la visión o el aprendizaje por refuerzo.

Pero no solo es una plataforma que ofrece servicio de hospedaje de modelos o la descarga de modelos, datasets, demos y otros contenidos de interés como documentación, sino que, más allá de todo ello, en el año 2023 lanzaron HuggingChat, un modelo que compite con ChatGPT basado en OpenAssistant,



Así, veamos diferentes opciones que ofrece el portal:

Prueba, explotación y publicación de modelos

Pinchando en el menú de la parte superior derecha, sobre la opción “Models” podemos acceder a un buscador de modelos impresionante, donde después de filtrar, por ejemplo, por tarea e idioma (panel izquierdo), se accede a una lista (panel derecho) de modelos coincidentes. En ese momento, puede pincharse sobre uno de ellos para acceder a la descripción del mismo y a su uso:

The screenshot shows two main sections of the Hugging Face Model Hub.

Left Panel (Models List):

- Header: Models 217, Filter by name, new Full-text search, Sort: Trending.
- Filter: Multimodal, Feature Extraction, Text-to-Image, Document Question Answering, Image-to-Video.
- Computer Vision: Image Classification, Image-to-Image, Object Detection, Unconditional Image Generation.
- Natural Language Processing: Text Classification, Text Generation, Translation, Token Classification, Text2Text Generation, Fill-Mask, Sentence Similarity, Question Answering, Summarization, Zero-Shot Classification, Conversational, Table Question Answering.

Right Panel (Model Card View):

- Model Card:** BSC-LT/roberta-base-bne-capitel-ner-plus
- YAML Metadata Errors:
 - YAML Metadata Error: "widget[0]" must be of type object
 - YAML Metadata Error: "widget[1]" must be of type object
- Description: Spanish RoBERTa-base trained on BNE finetuned for CAPITEL Named Entity Recognition (NER) dataset.
- Table of contents: Click to expand.
- Model description: The roberta-base-bne-capitel-ner-plus is a Named Entity Recognition (NER) model for the Spanish language fine-tuned from the roberta-base-bne model, a RoBERTa base model pre-trained using the largest Spanish corpus known to date, with a total of 570GB of clean and deduplicated text, processed for this work, compiled from the web crawlings performed by the National Library of Spain (Biblioteca Nacional de España) from 2009 to 2019. This model is a more robust version of the
- Files and versions: Downloads last month: 91, Inference API: Examples, Me llamo Andrés, nací en 1965 y vivo en la zona Sur de Madrid.
- Compute: Compute, Computation time on cpu: 0.062 s, Me llamo Andrés, nací en 1965 y vivo en la zona Sur LOC de Madrid. LOC, JSON Output, Maximize.
- Evaluation results: F1 on CAPITEL-NERC self-reported: 0.896, View on Papers With Code.

Desde la ficha de cualquier modelo se puede descargar (pestaña “Files and versions”), por lo que puedes explotarlo en local o, si lo prefieres, en la nube, desde Python y la librería Transformers de HuggingFace.

Descarga de datasets

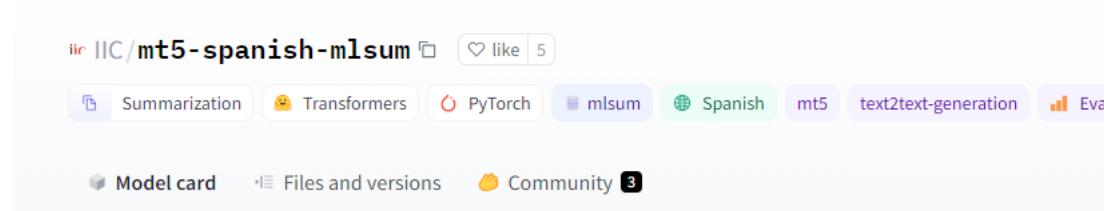
También puede accederse a un gran número de conjuntos de datos de entrenamiento.

Posts y docs

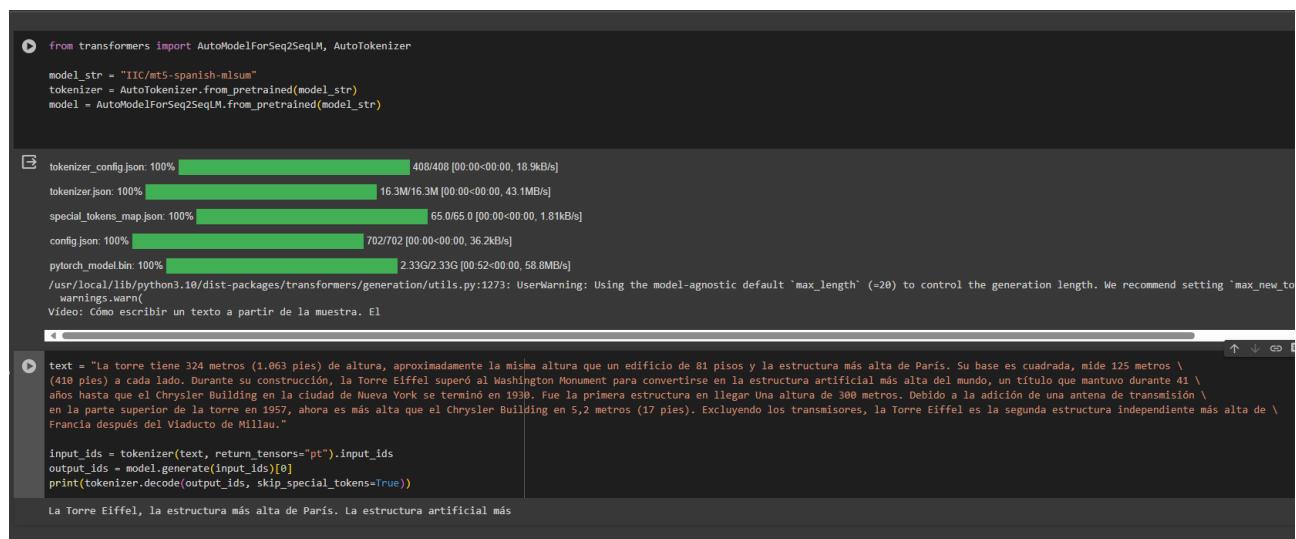
Si deseas mantenerte informado, en las secciones “Posts” y “Docs” verás publicaciones muy interesantes acerca de novedades y/o avances en materia de inteligencia artificial, por lo que te recomiendo que no pases por alto esta sección.

Ejemplo de uso

Probemos, por ejemplo, un modelo de sumarización.



El siguiente código ilustra cómo emplearlo en línea:



```
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer

model_str = "IIC/mt5-spanish-m1sum"
tokenizer = AutoTokenizer.from_pretrained(model_str)
model = AutoModelForSeq2SeqLM.from_pretrained(model_str)

tokenizer_config.json: 100% [██████████] 408/408 [00:00<00:00, 18.9kB/s]
tokenizer.json: 100% [██████████] 16.3M/16.3M [00:00<00:00, 43.1MB/s]
special_tokens_map.json: 100% [██████████] 65.0/65.0 [00:00<00:00, 1.81kB/s]
config.json: 100% [██████████] 702/702 [00:00<00:00, 36.2kB/s]
pytorch_model.bin: 100% [██████████] 2.33G/2.33G [00:52<00:00, 58.8MB/s]
/usr/local/lib/python3.10/dist-packages/transformers/generation/utils.py:1273: UserWarning: Using the model-agnostic default `max_length` (-20) to control the generation length. We recommend setting `max_new_tokens` instead.
warnings.warn(
Video: Cómo escribir un texto a partir de la muestra. El

text = "La torre tiene 324 metros (1.063 pies) de altura, aproximadamente la misma altura que un edificio de 81 pisos y la estructura más alta de París. Su base es cuadrada, mide 125 metros (410 pies) a cada lado. Durante su construcción, la Torre Eiffel superó al Washington Monument para convertirse en la estructura artificial más alta del mundo, un título que mantuvo durante 41 años hasta que el Chrysler Building en la ciudad de Nueva York se terminó en 1930. Fue la primera estructura en llegar Una altura de 300 metros. Debido a la adición de una antena de transmisión en la parte superior de la torre en 1957, ahora es más alta que el Chrysler Building en 5,2 metros (17 pies). Excluyendo los transmisores, la Torre Eiffel es la segunda estructura independiente más alta de Francia después del Viaducto de Millau."
input_ids = tokenizer(text, return_tensors="pt").input_ids
output_ids = model.generate(input_ids)[0]
print(tokenizer.decode(output_ids, skip_special_tokens=True))

La Torre Eiffel, la estructura más alta de París. La estructura artificial más
```

Puedes acceder al código de ejemplo desde la URL siguiente:

[jmolina010/ejemplo_hugging_face \(github.com\)](https://jmolina010/ejemplo_hugging_face)

Hugging Chat

En este caso, puedes acceder al modelo y probarlo en la dirección siguiente:

[HuggingChat \(huggingface.co\)](https://huggingchat.huggingface.co)

HuggingChat v0.7.0

Making the community's best AI chat models available to everyone.

Current Model
mistralai/Mixtral-8x7B-Instruct-v0.1



[Model page](#)

[Website](#)

Examples

[Write an email from bullet list](#)

[Code a snake game](#)

[Assist in a task](#)

Como puedes observar, hay un enlace al modelo en que está basado el chatbot.

Veamos, a continuación, un ejemplo de pregunta/respuesta:

describeme cómo debería desarrollarse un plan de negocio

- Desarrollar un plan de negocios es una tarea importante y necesaria si deseas iniciar tu propia empresa o hacer crecer un negocio existente. A continuación, te presento los pasos generales que debes seguir para crear un plan de negocios efectivo:
 1. Realiza una investigación de mercado: Antes de comenzar a escribir tu plan de negocios, es fundamental que realices una investigación de mercado exhaustiva. Esto incluye identificar a tu competencia directa e indirecta, conocer las preferencias y comportamientos de tus clientes potenciales, analizar tendencias del sector y determinar la demanda de tus productos o servicios.
 2. Define tu propuesta de valor única (PVU): Tu PVU es lo que distingue a tu negocio de la competencia y explica por qué tus clientes deben elegirte sobre otras opciones disponibles en el mercado. Debes ser capaz de expresar claramente tu PVU en pocas palabras.
 3. Crea un modelo de negocio: Describe cómo planeas obtener ingresos y mantener tu negocio funcionando. Incluye detalles como tu estructura de costos, precios de venta, canales de distribución y fuentes de financiamiento.
 4. Fija objetivos claros y medibles: Tus objetivos deben estar relacionados con el éxito a largo plazo de tu negocio. Deben ser específicos, alcanzables, relevantes y mensurables (objetivos SMART).
 5. Elabora un plan de marketing: Determina cómo llegarás a tus clientes potenciales y cómo promoverás tus productos o servicios. Considera diferentes canales de marketing, tales como redes sociales, publicidad en línea, eventos en vivo y correo electrónico.
 6. Planifica tus operaciones: Descubre cómo producirás y entregarás tus productos o servicios a tus clientes. Considera aspectos como recursos humanos, tecnología, procesos de fabricación y logística.
 7. Analiza los riesgos y establece mitigaciones: Identifica posibles riesgos que pueden afectar tu negocio, ya

Ejercicios propuestos

Ejercicio propuesto 1

Debes seleccionar un modelo y, basándote en el ejemplo del cuaderno, emplear el que hayas seleccionado para obtener un resultado visible.

Ejercicio propuesto 2

1. Explora las posibilidades conversacionales de HuggingChat,
2. Busca diferentes formas de comunicarte, investiga sobre ingeniería de prompting y experimenta con este chatbot.
3. Investiga acerca de hugchat (hugging chat API) y desarrolla un código de prueba. Nota: Empieza buscando en [Soulter/hugging-chat-api: HuggingChat Python API](#) (github.com)

Bibliografía para completar el aprendizaje

1. <https://huggingface.co/>
2. [jmolina010/ejemplo_hugging_face \(github.com\)](#)
3. [HuggingChat \(huggingface.co\)/chat](#)
4. [Soulter/hugging-chat-api: HuggingChat Python API](#) (github.com)