

Desarrollo de un Asistente Inteligente para la atención al cliente y solución técnica en empresas del sector del agua mediante modelos preentrenados

Grado: Ingeniería Informática

Trabajo Final Grado: Inteligencia Artificial

Autor: Antonio Fernández Salcedo

Consultor: Jonathan Ferrer Mestres

Profesor Responsable: Susana Acebo Nadal

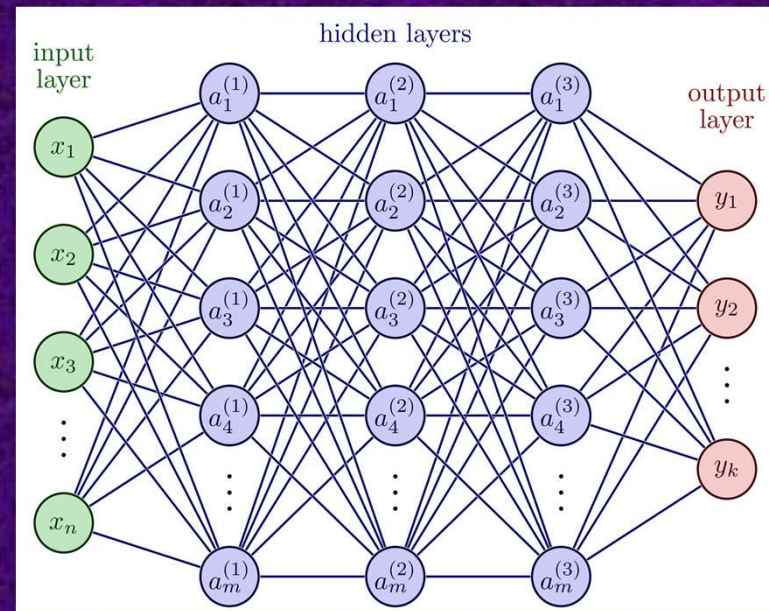
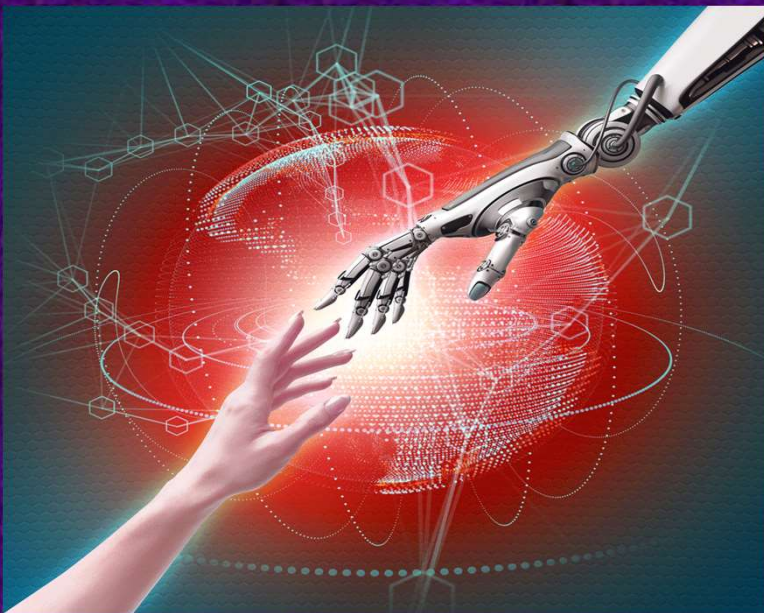
Contacto: afernandezsalc@uoc.edu

Unisersitat Oberta de Catalunya

Junio 2024

Presentación

El proyecto aborda el desarrollo y evaluación de un asistente inteligente basado en modelos preentrenados de lenguaje natural, con el objetivo de crear un sistema capaz de ofrecer respuestas precisas y mejorar la atención a los usuarios de la empresa Canal de Isabel II mediante técnicas de inteligencia artificial y *Natural Language Processing*.



Resumen del contenido del Trabajo Final Grado

1. Introducción.
2. Objetivos.
3. Análisis.
4. Diseño.
5. Implementación.
6. Evaluación.
7. Productos obtenidos.
8. Trabajos futuros.
9. Conclusiones.

1. Introducción

Este proyecto se enfoca en la implementación de modelos preentrenados de lenguaje, como BERT o Mixtral, para llevar a cabo *fine-tuning* e inferencias y desarrollar un asistente inteligente. Este asistente está destinado a proporcionar atención al cliente y ofrecer soluciones técnicas específicas para empresas del sector del agua, tomando como referencia el Canal de Isabel II.

La metodología del proyecto sigue un enfoque en cascada, incorporando técnicas de prototipado y pruebas tempranas, con la recolección progresiva y el análisis continuo de un conjunto de datos de preguntas y respuestas etiquetadas del Canal de Isabel II.



Fuente: <https://medium.com>

2. Objetivos

El **objetivo principal** es:

- 1. Desarrollar un asistente inteligente diseñado específicamente para empresas del sector del agua, con un enfoque particular en el Canal de Isabel II.

- 3. Realizar *fine-tuning* e inferencias en un modelo preentrenado y adaptarlo al contexto del Canal de Isabel II.

Objetivos específicos:

- 2. Investigar la aplicabilidad de modelos preentrenados.

- 4. Diseñar la interfaz del asistente para facilitar la interacción con los clientes y técnicos del Canal de Isabel II.

3. Análisis

Revisar la literatura existente sobre el uso de modelos preentrenados.

El uso de modelos preentrenados como BERT en aplicaciones de Procesamiento del Lenguaje Natural (PLN), como los sistemas de pregunta-respuesta, ha revolucionado la forma en que representamos el significado de las palabras y realizamos tareas de PLN de alta complejidad.

Utilizaremos BERT como base para nuestro modelo. BERT (*Bidirectional Encoder Representations from Transformers*) es un modelo de lenguaje preentrenado que ha demostrado un rendimiento excepcional en una variedad de tareas de procesamiento del lenguaje natural. Emplearemos un modelo preentrenado de BERT y lo adaptaremos para realizar la tarea de Q&A

- Modelo BERT con TensorFlow.
- Modelo BERT preentrenado con PyTorch.
- Modelo DistilBERT en idioma inglés .
- Modelo BERT preentrenado (*fine-tuned*).
- Uso de la función *pipeline* para simplificar el proceso de cargar un modelo preentrenado, tokenizar datos de entrada y ejecutar inferencias.

Los modelos de respuesta a preguntas pueden recuperar la respuesta a una pregunta de un texto determinado, lo que resulta útil para buscar una respuesta en un documento. Algunos modelos de respuesta a preguntas pueden generar respuestas sin contexto.

Análisis de modelos preentrenados para *fine-tuning*.

Recopilación y análisis del conjunto de datos.

El conjunto de datos servirá como base para entrenar los modelos y ajustarlos a las necesidades específicas de la empresa.

Los datos etiquetados que utilizaremos contienen información relevante de la oficina virtual del Canal de Isabel II (averías, facturación, bonificaciones, consumos, lecturas, formas de pago, contratación, cambio de titularidad, gestiones, contratos y contacto), específicamente diseñados para resolver consultas de los usuarios. El dataset está en formato JSON.

4. Diseño

4.1. Seleccionar modelos preentrenados adecuados para *fine-tuning*



En el proyecto se sigue una estrategia de diseño basada en la adaptación de varios modelos preentrenados: Mixtral y BERT.

Se selecciona el modelo BERT por su desempeño en comprensión del lenguaje natural y la variedad de versiones, y se realiza su ajuste (*fine-tuning*) al dominio del sector del agua.



El tamaño del modelo BERT puede variar significativamente y puede afectar el rendimiento y la eficiencia del modelo.

Es crucial elegir un modelo BERT que esté preentrenado en el idioma que se necesita, que es español.

Antes de elegir un modelo BERT para *fine-tuning*, es importante evaluar la calidad de las respuestas que proporciona en pruebas de evaluación relevantes.

Hay que considerar si el modelo se adapta mejor a la tarea específica de preguntas y respuestas .



Al elegir entre diferentes modelos BERT para *fine-tuning*, hay varias opciones que considerar para tomar la mejor decisión:

- a. Tamaño del modelo.
- b. Idioma/s con que se preentrenó el modelo.
- c. Calidad de las respuestas de prueba.
- d. Compatibilidad con la tarea específica de preguntas y respuestas.



Al evaluar estas opciones y consideraciones, podemos seleccionar el modelo BERT más adecuado para el proyecto de *fine-tuning* y obtener resultados óptimos en las tareas de procesamiento del lenguaje natural. El modelo seleccionado finalmente es:

[dccuchile/bert-base-spanish-wwm-uncased-finetuned-ga-mlqa](https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased-finetuned-ga-mlqa) · Hugging Face

4. Diseño

4.2. Definir los parámetros y características específicas del *fine-tuning* para adaptar el modelo al contexto del Canal de Isabel II



1. *Fine-tuning* supervisado: Utilizar el conjunto de datos etiquetado de preguntas y respuestas para ajustar los pesos del modelo específicamente para esta tarea. Dado que el conjunto de datos es pequeño, esta técnica puede ser efectiva para adaptar el modelo a la tarea específica.



2. *Transfer learning*: Aprovechar el conocimiento preentrenado del modelo BERT en el modelado del lenguaje español. Utilizar el modelo preentrenado como punto de partida y ajustar los pesos del modelo para la tarea de preguntas y respuestas utilizando el conjunto de datos etiquetado.



3. Ajuste de hiperparámetros: Experimentar con diferentes valores de hiperparámetros, como la tasa de aprendizaje, el tamaño del lote, la función de pérdida y el número de épocas de entrenamiento. Realizar una búsqueda exhaustiva de hiperparámetros para encontrar la combinación óptima que maximice el rendimiento del modelo en el conjunto de datos.

4. *Data augmentation*: Dado que el conjunto de datos es relativamente pequeño, se considera aplicar técnicas de aumento de datos para aumentar su tamaño y mejorar la capacidad de generalización del modelo. En este caso se aumentará el tamaño del conjunto de datos mediante la reformulación de preguntas y manteniendo el contexto que es una estrategia efectiva.



5. Afinación por capas: Se agregan capas adicionales al final del clasificador de salida del modelo BERT, con ello estamos extendiendo la arquitectura del modelo al añadir más capas neuronales justo antes de las capas finales que predicen las respuestas, además permite al modelo aprender representaciones más complejas. También se pueden congelar algunas capas del modelo BERT.

4. Diseño

4.3. Seleccionar un modelo para ajustar parámetros de generación de texto para realizar inferencias específicas en el contexto Canal de Isabel II



Los modelos abiertos de Mixtral establecen el estándar de eficiencia y están disponibles de forma gratuita, con licencia totalmente permisiva.



Para decidir qué modelo es mejor para hacer inferencias, se deben considerar las necesidades específicas y el tipo de datos con los que trabajaremos. Algunos puntos que considerar:

1. Especialización en instrucciones.
2. Generalización.
3. Pruebas y experimentación.



El modelo Mistral 7B es de despliegue rápido y fácilmente personalizable. Los modelos Mistral 8x7B se han optimizado a través de un ajuste fino supervisado y una optimización directa de preferencias (DPO) para un seguimiento cuidadoso de las instrucciones.

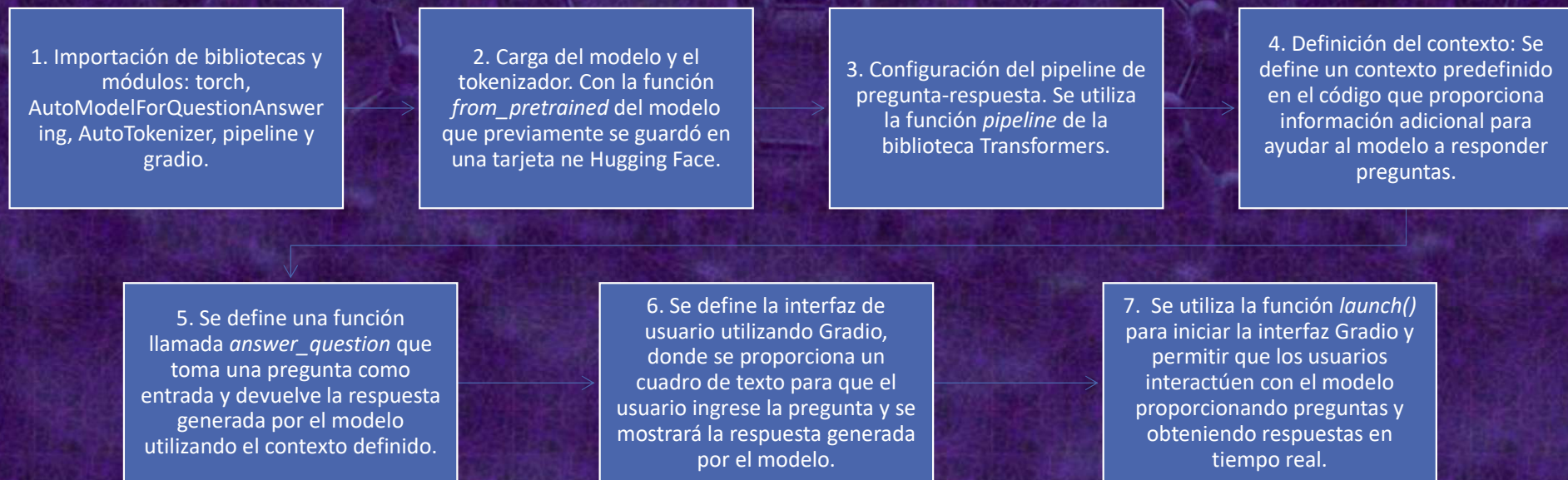


Para realizar las inferencias se elige el modelo Mistral-8x7B-Instruct-v0.1 debido a su capacidad para seguir instrucciones específicas y guiar a los clientes o técnicos del Canal de Isabel II en la resolución de problemas o la realización de tareas.

5. Implementación

5.1. Implementar el asistente inteligente con el modelo BERT preentrenado ajustado

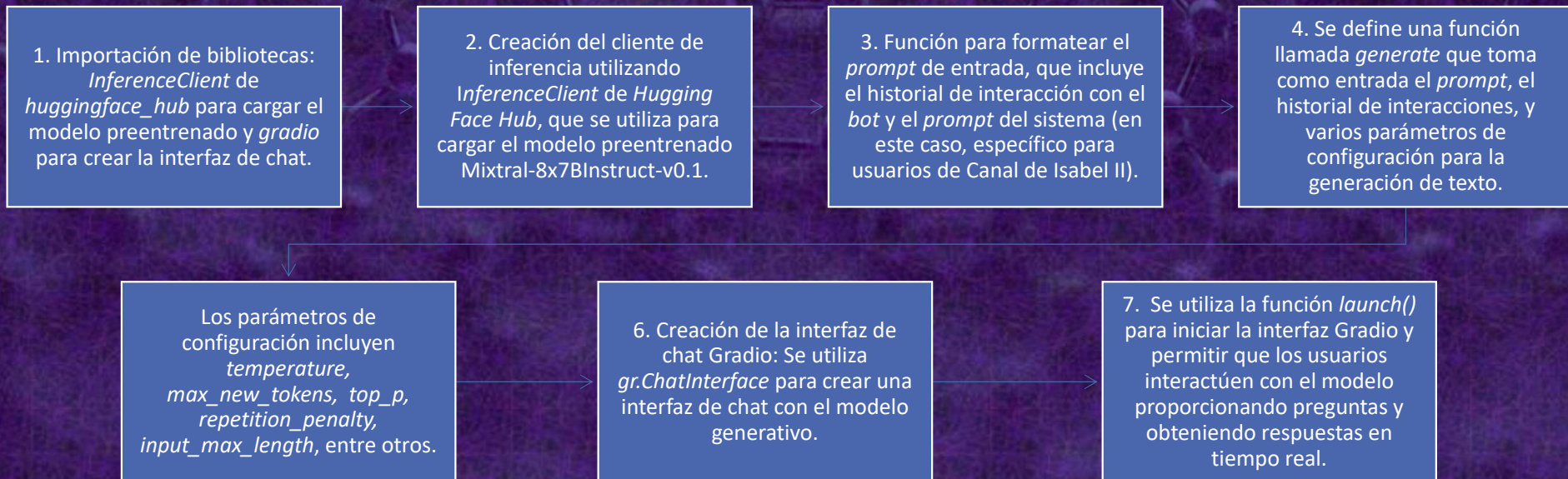
El despliegue del modelo BERT preentrenado con *fine-tuning* específico para preguntas y respuestas utilizando el *framework* Hugging Face, se concentra en el archivo `app.py`. Este archivo carga el modelo BERT *finetuned* para preguntas y respuestas y proporciona una interfaz de usuario simple para interactuar con el modelo y obtener respuestas en función de las preguntas ingresadas. Se detallan los componentes y funciones clave del archivo:



5. Implementación

5.2. Diseñar la interfaz del asistente con el modelo Mixtral para facilitar la interacción con los clientes o técnicos del Canal de Isabel II

El despliegue del modelo Mixtral preentrenado, se concentra en el archivo `app.py`. Este archivo carga un modelo Mixtral preentrenado para realizar inferencias, define funciones para formatear el *prompt* y generar respuestas, y crea una interfaz de chat interactiva utilizando Gradio para que los usuarios de la empresa Canal de Isabel II puedan interactuar con el modelo y obtener respuestas en tiempo real. La explicación detallada del archivo:



6. Evaluación

En la evaluación del rendimiento de los modelos se utilizan diversas métricas léxicas:

- Precision: (Palabras coincidentes / Total palabras respuesta generada)
- Recall: (Palabras coincidentes / Total palabras respuesta referencia)
- F1-Score: $2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$
- Accuracy: (Respuestas correctas / Total respuestas)

Se realizan entrenamientos del modelo con los mismos parámetros, utilizando diferentes optimizadores: *SGD*, *Adam* y *AdamW*. Se espera que un buen entrenamiento se traduzca en respuestas de calidad y, por lo tanto, en una evaluación favorable.

Se implementa un filtrado de respuestas basado en el *score* de confianza, eliminando aquellas con un *score* inferior a cierto umbral. Este enfoque pretende mejorar la calidad de las respuestas al descartar aquellas que el modelo considera menos confiables.

En general, los resultados obtenidos sugieren que el modelo puede beneficiarse de mejoras en su capacidad para generar respuestas correctas. Es importante revisar el proceso de generación de respuestas, así como la calidad y cantidad de datos utilizados para el entrenamiento, para identificar áreas específicas que requieran ajustes y mejoras.

6. Evaluación

Identificar áreas de mejora

Se puede experimentar con diferentes enfoques y ajustes para encontrar la configuración que mejor se adapte a las necesidades y requisitos de rendimiento. Se proponen:



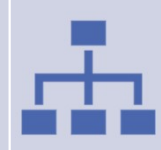
Regularización: Para evitar el sobreajuste en un conjunto de datos pequeño, se puede considerar aplicar técnicas de regularización como la *dropout* durante el entrenamiento. La *dropout* puede ayudar a reducir la dependencia entre las neuronas y mejorar la generalización del modelo.



Ampliar el dataset: Aumentar el tamaño y la diversidad del conjunto de datos puede ayudar al modelo a capturar mejor la variabilidad del lenguaje y mejorar su capacidad para generar respuestas precisas y relevantes.



Afinación por capas: Dado el tamaño relativamente pequeño del conjunto de datos, se pueden considerar congelar algunas capas del modelo preentrenado y solo ajustar los parámetros de las capas superiores durante el *fine-tuning*. Esto puede ayudar a evitar el sobreajuste y mejorar la eficiencia del entrenamiento.



Experimentar con arquitecturas de modelos alternativos: Probar diferentes arquitecturas de modelos, como modelos más grandes o modelos específicos diseñados para tareas de generación de texto, podría ayudar a mejorar el rendimiento del modelo en la generación de respuestas.

7. Productos obtenidos

- ❑ Preguntas y respuestas a usuarios utilizando el modelo BERT con Fine-Tuning.
 - <https://antonio49-bertfinetuningcontextopredefinido2.hf.space>
 - <https://github.com/antoniodeepblue/CanalBERTfine-tuning>
 - <https://huggingface.co/spaces/Antonio49/BERTfinetuningContextoPredefinido2>

- ❑ Preguntas y respuestas a usuarios utilizando el modelo Mixtral mediante inferencia.
 - <https://antonio49-chat-mixtral-aguausuarioconhilos.hf.space>
 - <https://github.com/antoniodeepblue/Chat.Mixtral.AguaUsuarioConHilos>
 - <https://huggingface.co/spaces/Antonio49/Chat.Mixtral.AguaUsuarioConHilos>

- ❑ Asistencia de experto en el sector del agua utilizando el modelo Mixtral mediante inferencia.
 - <https://antonio49-chat-mixtral-expertoconhilos.hf.space>
 - <https://github.com/antoniodeepblue/Chat.Mixtral.ExpertoConHilos>
 - <https://huggingface.co/spaces/Antonio49/Chat.Mixtral.ExpertoConHilos>

- ❑ Tarjeta modelo BERT con Fine-Tuning (modelo guardado):
 - <https://github.com/antoniodeepblue/ModeloCanalBERT>
 - <https://huggingface.co/Antonio49/ModeloCanal>

- ❑ Carpeta principal del Trabajo Final Grado:
 - <https://drive.google.com/drive/folders/1U3TzPPWYLWy1MWekZ5t0nUU58gJN3HMo>

8. Trabajos futuros



1. Exploración de técnicas avanzadas de *fine-tuning*: Implementar técnicas como la regularización o la afinación (congelación y/o agregación) por capas para mejorar el rendimiento del modelo y evitar el sobreajuste.



2. Ampliación y diversificación del conjunto de datos: Incorporar más datos de diferentes áreas del sector del agua para mejorar la generalización del modelo.



3. Evaluación y exploración de otros modelos preentrenados: Investigar modelos adicionales como Mixtral y su aplicabilidad en el contexto del proyecto.



4. Exploración de otras aplicaciones: Investigar la aplicabilidad del asistente inteligente en otros sectores o ámbitos.



5. Integración con sistemas existentes: Integrar el asistente inteligente con los sistemas informáticos existentes para una mayor eficiencia y accesibilidad.



6. Evaluación continua y ajustes: Evaluar el rendimiento del asistente a lo largo del tiempo y realizar ajustes periódicos para mantener su eficacia.

9. Conclusiones

En este proyecto, la planificación detallada al inicio ha sido fundamental para el éxito de este, incluyendo una distribución eficiente de tareas y la definición de hitos claros que han permitido un desarrollo ordenado. Por un lado, la selección y análisis del conjunto de datos resultaron cruciales para la eficacia del modelo *fine-tuning*. Por otro lado, la investigación de modelos preentrenados de lenguaje natural me permitió elegir el más adecuado para nuestras necesidades.

En cuanto a los objetivos establecidos, se han alcanzado en su totalidad: se realizó un estudio profundo de modelos preentrenados, se personalizó BERT para el sector del agua y se desarrolló un asistente inteligente que destaca por su eficiencia y precisión. Además, el nivel de dificultad del proyecto está en perfecta armonía con las habilidades que se requieren en mi formación académica.

Cabe mencionar que la memoria junto al Notebook y los anexos sirven como una guía completa para entender cada fase del desarrollo, el proceso de ajuste fino, la inferencia del asistente y la evaluación del modelo. También se incluyen explicaciones pormenorizadas y el código fuente comentado para facilitar su comprensión y aplicación. Este código es modular y reutilizable, permitiendo así que otros investigadores o desarrolladores lo adapten según sus propios requerimientos.

El proyecto representó un reto bien alineado con mis competencias y a nivel personal quiero destacar que me ha ofrecido la oportunidad de desarrollar importantes habilidades, como la capacidad de trabajar de forma autónoma, investigar y comprender la aplicabilidad de los modelos preentrenados, desarrollar una capacidad de análisis y resolución de problemas, entre otras.

En conclusión, el proyecto sienta las bases para un asistente inteligente efectivo en la atención a usuarios del Canal de Isabel II. Además, he delineado una serie de trabajos futuros que incluyen ajustes y mejoras adicionales para maximizar la precisión y la calidad de las interacciones.

Fuentes bibliográficas utilizadas 1

- ❖ Joaquim Moré López, Introducción al procesamiento del lenguaje natural. PID_00287316, 1ª Edición, Fundació Universitat Oberta de Catalunya (FUOC), Barcelona, 2022.
- ❖ Smaller BERT Models. [en línea] [consulta: 1 de abril de 2024]. Disponible en: <https://github.com/google-research/bert>
- ❖ BERT and Transfer Learning in NLP. [en línea] [consulta: 1 de abril de 2024]. Disponible en: <https://medium.com/@mervebdurna/bert-and-transfer-learning-in-nlp-11fc19435fa0>
- ❖ Fine-Tuning BERT for text classification with LoRA. [en línea] [consulta: 2 de abril de 2024]. Disponible en: <https://medium.com/@karkar.nizar/fine-tuning-bert-for-text-classification-with-lora-f12af7fa95e4>
- ❖ BERT para responder preguntas sobre SQuAD 2.0 en español. [en línea] [consulta: 3 de abril de 2024]. Disponible en: <https://jaimesendraberenguer.medium.com/bert-para-responder-preguntas-sobre-squad-2-0-en-español-5842748f051a>
- ❖ SQUAD_es_v8_GPU.ipynb. [en línea] [consulta: 4 de abril de 2024]. Disponible en: https://colab.research.google.com/github/jaisenbe58r/iAApi-QAS-BERT/blob/main/SQUAD_es_GPU.ipynb
- ❖ Saca el máximo partido de tu suscripción a Colab. [en línea] [consulta: 4 de abril de 2024]. Disponible en: <https://colab.research.google.com/notebooks/pro.ipynb>
- ❖ Mastering BERT: A Comprehensive Guide from Beginner to Advanced in Natural Language Processing (NLP). [en línea] [consulta: 5 de abril de 2024]. Disponible en: <https://medium.com/@shaikhrayyan123/a-comprehensive-guide-to-understanding-bert-from-beginners-to-advanced-2379699e2b51>
- ❖ Analyzing Semantic Equivalence of Sentences Using BERT. [en línea] [consulta: 6 de abril de 2024]. Disponible en: <https://www.analyticsvidhya.com/blog/2022/02/analyzing-semantic-equivalence-of-sentences-using-bert/>
- ❖ Clasificar texto con BERT. [en línea] [consulta: 7 de abril de 2024]. Disponible en: https://www.tensorflow.org/text/tutorials/classify_text_with_bert?hl=es-419
- ❖ Fine-tuning a BERT model. [en línea] [consulta: 8 de abril de 2024]. Disponible en: https://www.tensorflow.org/tfmodels/nlp/fine_tune_bert
- ❖ Recuperación de preguntas y respuestas del codificador de oraciones universal multilingüe. [en línea] [consulta: 9 de abril de 2024]. Disponible en: https://www.tensorflow.org/hub/tutorials/retrieval_with_tf_hub_universal_encoder_qa?hl=es-419
- ❖ Ajuste fino con conjuntos de datos personalizados. [en línea] [consulta: 10 de abril de 2024]. Disponible en: https://huggingface.co/transformers/v3.3.1/custom_datasets.html

Fuentes bibliográficas utilizadas 2

- ❖ Ajuste fino de BERT para preguntas y respuestas. [en línea] [consulta: 10 de abril de 2024]. Disponible en: https://huggingface.co/docs/transformers/tasks/question_answering
- ❖ Ajuste fino de RoBERTa para preguntas y respuestas. [en línea] [consulta: 10 de abril de 2024]. Disponible en: <https://huggingface.co/mrm8488/roberta-base-1B-1-finetuned-squadv1>
- ❖ QLoRa: Ajuste un modelo de lenguaje grande en su GPU. [en línea] [consulta: 11 de abril de 2024]. Disponible en: <https://kaitchup.substack.com/p/qlora-fine-tune-a-large-language-model-on-your-gpu-27bed5a03e2b>
- ❖ PEFT (Parameter-Efficient Fine-Tuning). Hugging Face. [en línea] [consulta: 11 de abril de 2024]. Disponible en: <https://huggingface.co/docs/peft/main/en/index>
- ❖ Ajuste fino del modelo de lenguaje cuantificado usando LoRA con peft y transformadores en GPU T4. [en línea] [consulta: 10 de abril de 2024]. Disponible en: <https://towardsdev.com/fine-tune-quantized-language-model-using-lora-with-peft-transformers-on-t4-gpu-287da2d5d7f1>
- ❖ QLoRA: ajuste eficiente de LLM cuantificados. [en línea] [consulta: 11 de abril de 2024]. Disponible en: <https://github.com/artidoro/qlora>
- ❖ LoRA y QLoRA: métodos efectivos para afinar sus LLM en detalle. [en línea] [consulta: 12 de abril de 2024]. Disponible en: <https://medium.com/@levxn/lora-and-qlora-effective-methods-to-fine-tune-your-llms-in-detail-6e56a2a13f3c>
- ❖ SomosNLP. [en línea] [consulta: 12 de abril de 2024]. Disponible en: https://somosnlp.org/recursos/tutoriales/07_entrenamiento_finetuning
- ❖ Welcome Gemma - Google's new open LLM. [en línea] [consulta: 13 de abril de 2024]. Disponible en: <https://huggingface.co/blog/gemma>
- ❖ HuggingChat. [en línea] [consulta: 13 de abril de 2024]. Disponible en: <https://huggingface.co/chat/>
- ❖ Hugging Face. About Question Answering. [en línea] [consulta: 14 de abril de 2024]. Disponible en: <https://huggingface.co/tasks/question-answering>
- ❖ Canal de Isabel II. [en línea] [consulta: 21 de abril de 2024]. Disponible en: <https://www.canaldeisabelsegunda.es>
- ❖ Canal de Isabel II, oficina virtual. [en línea] [consulta: 22 de abril de 2024]. Disponible en: <https://oficinavirtual.canaldeisabelsegunda.es>
- ❖ Repositorio de modelos de HuggingFace. [en línea] [consulta: 23 de abril de 2024]. Disponible en: <https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased>
- ❖ BETO: Español BERT. [en línea] [consulta: 24 de abril de 2024]. Disponible en: <https://github.com/dccuchile/beto>
- ❖ DCC UChile. Departamento de Ciencias de la Computación Universidad de Chile. [en línea] [consulta: 25 de abril de 2024]. Disponible en: <https://github.com/dccuchile>

Fuentes bibliográficas utilizadas 3

- ❖ Mistral IA generativa abierta y portátil para desarrolladores y empresas. [en línea] [consulta: 30 de abril de 2024]. Disponible en: <https://mistral.ai>
- ❖ Mistral mistralai/Mixtral-8x7B-Instruct-v0.1. [en línea] [consulta: 30 de abril de 2024]. Disponible en: <https://huggingface.co/mistralai>
- ❖ Mistral Transformer. [en línea] [consulta: 30 de abril de 2024]. Disponible en: <https://github.com/mistralai/mistral-src>
- ❖ Mixtral-8x7B: Understanding and Running the Sparse Mixture of Experts by Mistral AI. [en línea] [consulta: 1 de mayo de 2024]. Disponible en: <https://kaitchup.substack.com/p/mixtral-8x7b-understanding-and-running>
- ❖ (Parte 1) Construye tu propio RAG con Mistral-7B y LangChain. [en línea] [consulta: 2 de mayo de 2024]. Disponible en: <https://medium.com/@thakermadhav/build-your-own-rag-with-mistral-7b-and-langchain-97d0c92fa146>
- ❖ (Parte 2) Construye tu propio RAG con Mistral-7B y LangChain. [en línea] [consulta: 2 de mayo de 2024]. Disponible en: <https://medium.com/@thakermadhav/part-2-build-a-conversational-rag-with-langchain-and-mistral-7b-6a4ebe497185>
- ❖ Modelos preentrenados Kaggle. [en línea] [consulta: 3 de mayo de 2024]. Disponible en: <https://www.kaggle.com/models?query=text-embedding&tfhub-redirect=true>
- ❖ Encontrar el equilibrio adecuado entre el uso de la memoria y la velocidad de inferencia. [en línea] [consulta: 3 de mayo de 2024]. Disponible en: <https://towardsdatascience.com/run-mixtral-8x7b-on-consumer-hardware-with-expert-offloading-bd3ada394688>
- ❖ Bibliotecas:
 - Transformers: <https://github.com/huggingface/transformers>
 - TensorFlow: <https://www.tensorflow.org/>
 - PyTorch: <https://pytorch.org/>
- ❖ Uso de herramientas de inteligencia artificial generativa (IAG): Modelos preentrenados para diferentes aplicaciones dentro del ámbito del NLP y la generación de texto que se han utilizado para investigar, buscar y contrastar información, conseguir puntos clave, traducir textos, etc. referente al TFG:
 - ChatGPT (OpenAI). <https://chat.openai.com>
 - Mixtral. <https://mistral.ai>
 - Gemini (Google). <https://gemini.google.com>
 - Copilot. <https://www.microsoft.com/es-es/edge/features/copilot?form=MT00D8>

Antonio Fernández Salcedo le agradece el interés
mostrado en el trabajo y espero que le resulte
interesante el contenido. Saludos.



Fuente: <https://www.its-security.es/plataforma/>

Universitat Oberta de Catalunya
Trabajo Final Grado: Inteligencia Artificial