

# Generación de Contenido Autoayuda y Apoyo Psicológico Personalizado



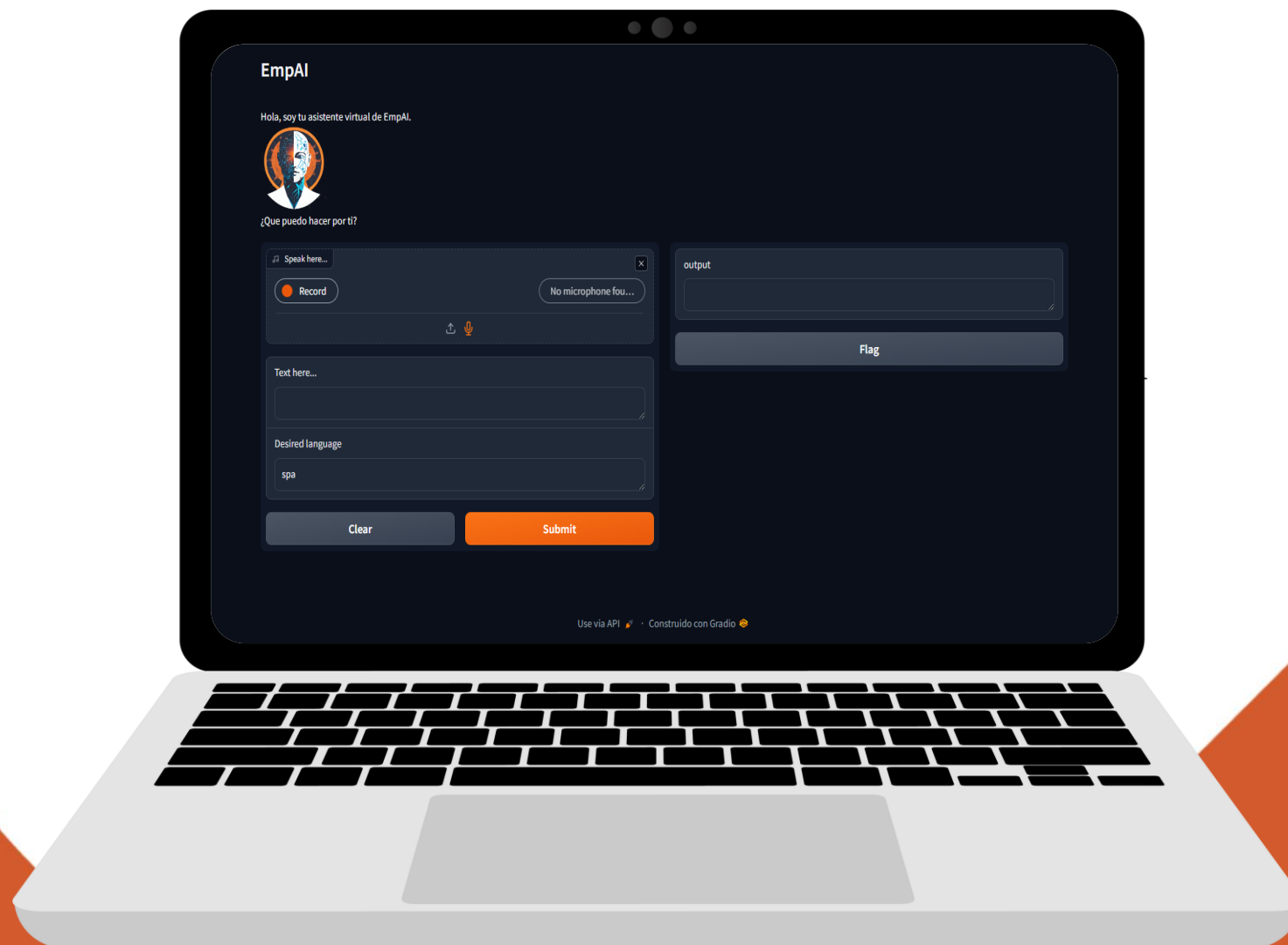
**Universidad**  
Internacional  
de Valencia

TFM – Máster en Inteligencia Artificial

Alumno: Motta Valero Luis Angel

Tutor: Cigales Canga Jesús

# EmpAI



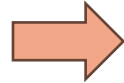
# INDICE DE CONTENIDO

- Introducción
- Marco Teórico
- Estado del Arte
- Material
- Métodos
- Resultados
- Conclusiones
- Próxima investigación

# Introducción

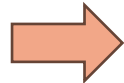
En la sociedad contemporánea, el apoyo psicológico emerge como una necesidad vital, dada la creciente importancia del bienestar emocional. En el contexto actual, donde el bienestar emocional cobra una relevancia cada vez mayor, la demanda de sistemas de apoyo psicológico personalizado se ha incrementado significativamente.

Médico-psiquiatra Marian Rojas  
Estapé (2018, pág. 159)



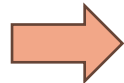
La acumulación de emociones reprimidas por la falta de aceptación y relevancia, en algún momento puede desencadenar enfermedades psicosomáticas.

Schütz Balistieri & Mara de Melo  
Tavares (2013)



Su estudio demuestra que el respaldo emocional juega un papel crucial durante períodos de salud precaria.

Rafael Santana Mondragón  
(2017)



El acto médico de un especialista en psicología va más allá del tratamiento clínico y abraza la empatía como herramienta para aliviar cansancios.

**Objetivo:** Contribuir al mejoramiento del estado emocional del usuario mediante respuestas empáticas desde las diferentes modalidades que adopta el flujo conversacional entre las personas, dentro de las que incluye la modalidad lingüística y acústica.

**Propuesta:** Creación de un sistema de conversación innovador que integre diversas tecnologías para brindar un soporte integral a los usuarios.

Generación de Respuestas Empáticas

Análisis Multimodal de Sentimientos

Traducción Multimodal

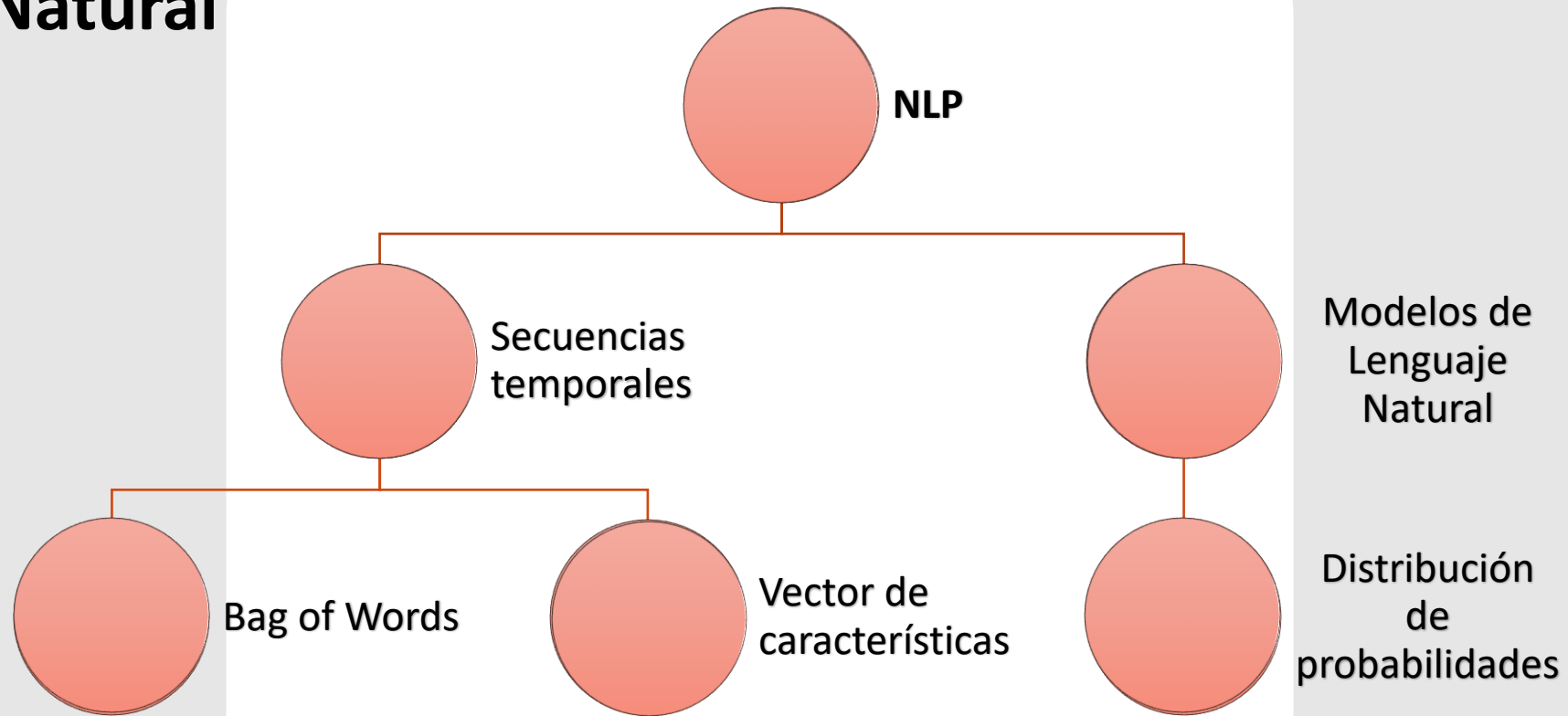
# INDICE DE CONTENIDO

- Introducción
- Marco Teórico
- Estado del Arte
- Material
- Métodos
- Resultados
- Conclusiones
- Próxima investigación

# Marco Teórico



# Procesamiento de Lenguaje Natural – NLP

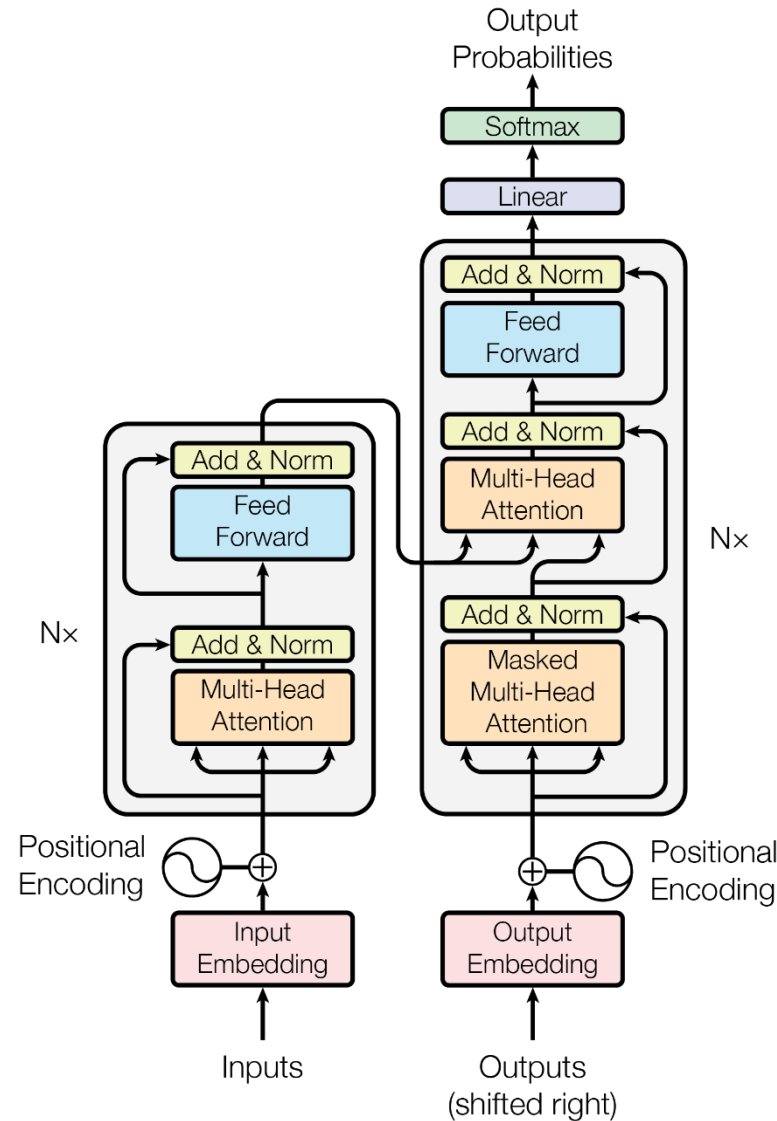


# Modelo Transformers

Arquitectura con mecanismos de autoatención

Codificadores

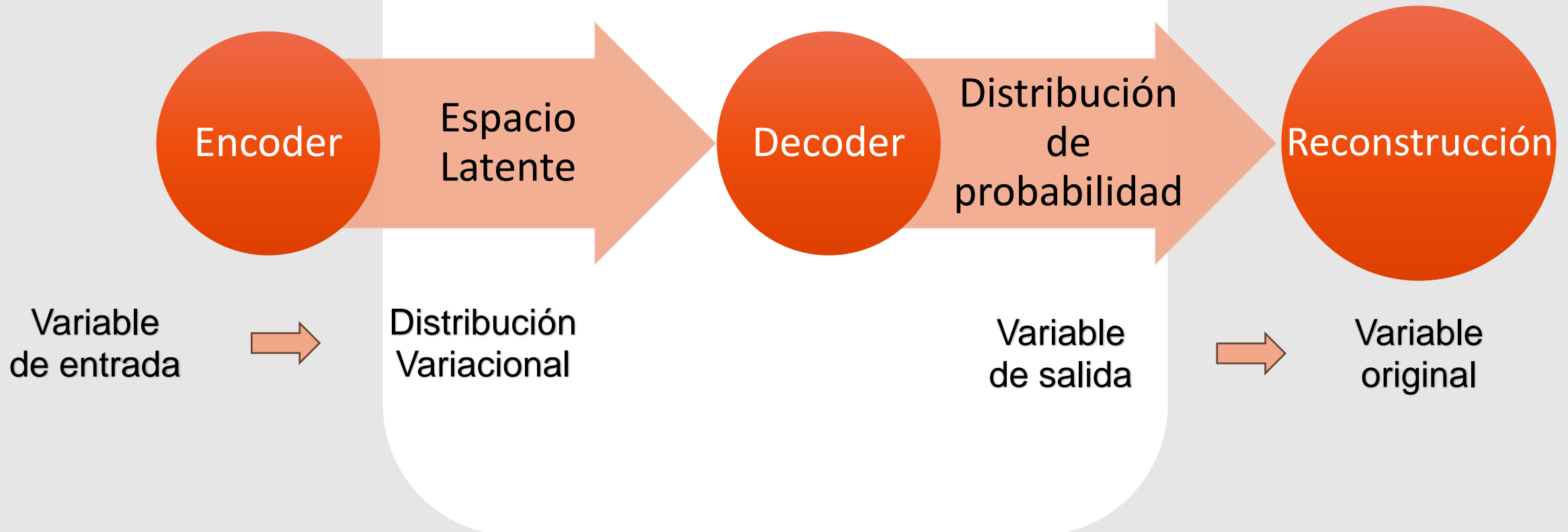
Capa de autoatención y una red neuronal



Decodificadores

Capa de autoatención.  
Capa de atención codificador-decodificador.  
Una red neuronal.

# AutoEncoder Variacional – VAE



# Modelos de Lenguaje

- Generación de texto
- Asistentes de codificación inteligente
- Predecir el siguiente token
- Acceso a los tokens previos

Causal

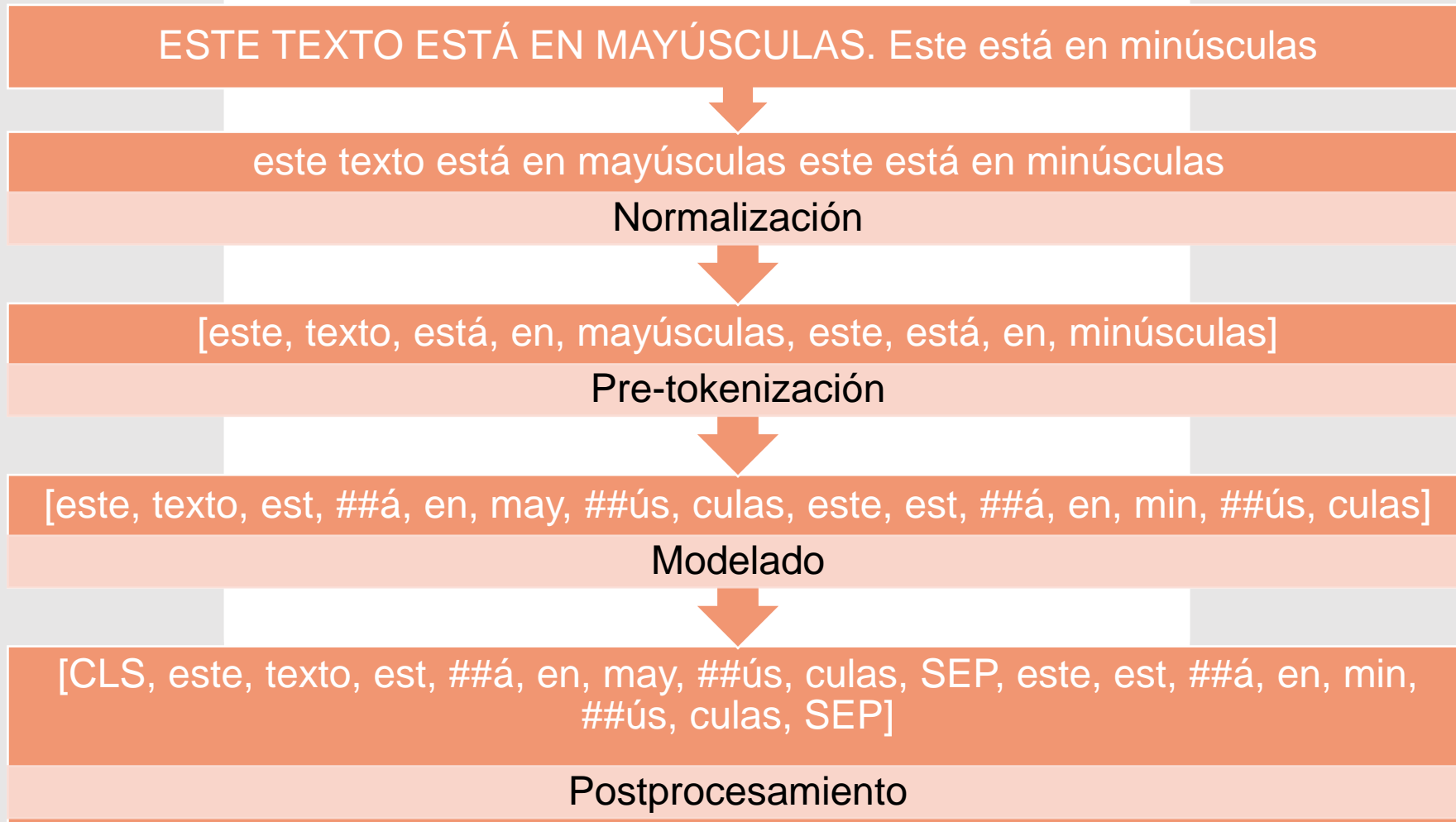
GPT-2

- Comprensión contextual
- Predecir tokens enmascarados
- Atención a ambas direcciones

Enmascarado

RoBERTa

# Preprocesamiento en el Modelado de Lenguaje



# Data Collator

## DataCollatorWithPadding

Completa dinámicamente las entradas recibidas mediante la adición de relleno según sea necesario.

## DataCollatorForLanguageModeling

Rellena dinámicamente las entradas hasta la longitud máxima de un lote si estas no tienen la misma longitud. Se utiliza específicamente para el modelado de lenguaje.

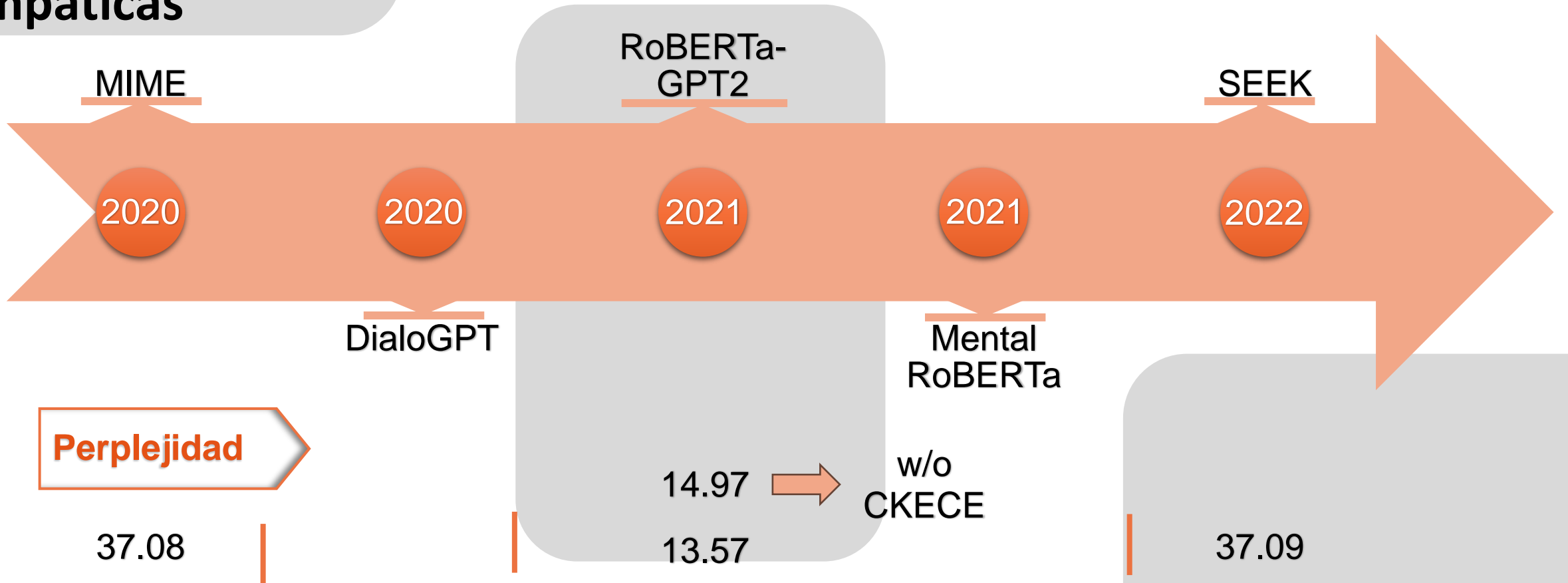
# INDICE DE CONTENIDO

- Introducción
- Marco Teórico
- Estado del Arte
- Material
- Métodos
- Resultados
- Conclusiones
- Próxima investigación

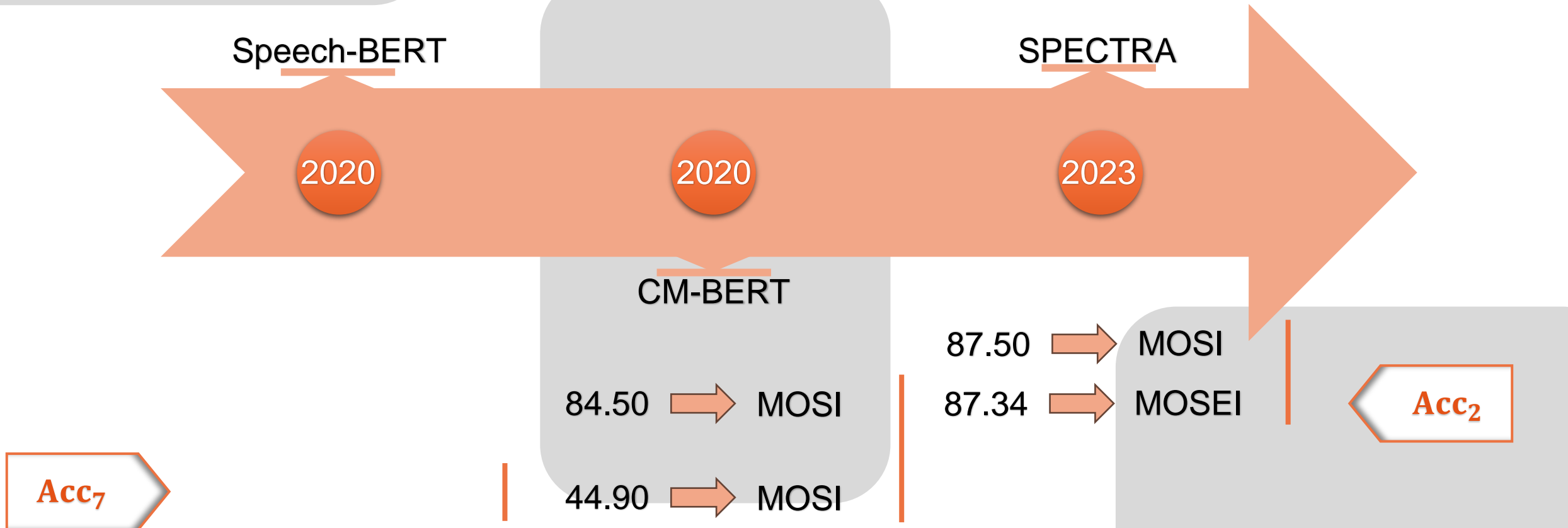
# Estado del Arte



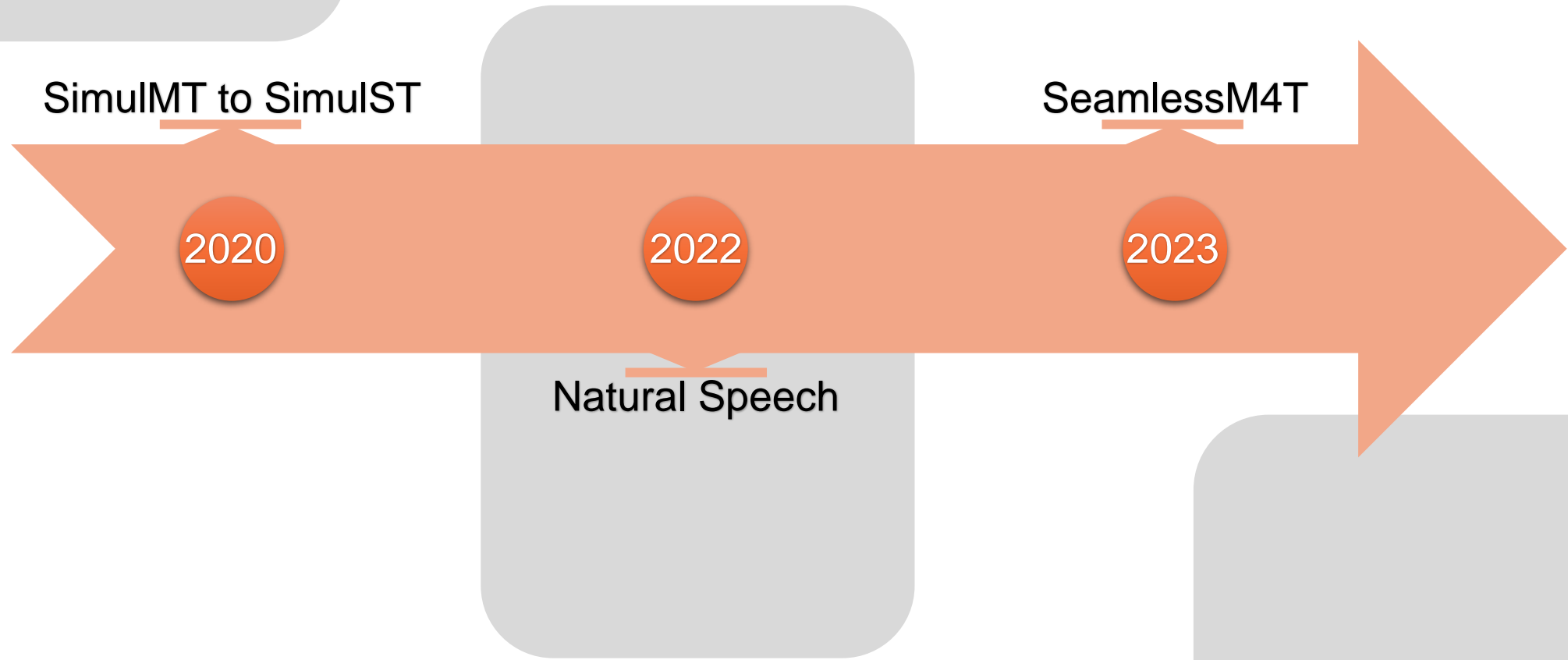
# Generación de Respuestas Empáticas



# Análisis Multimodal de Sentimientos



# Traducción Multimodal

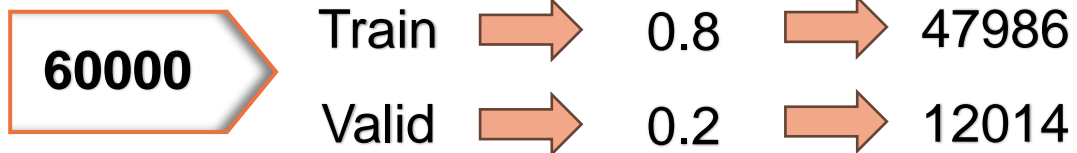
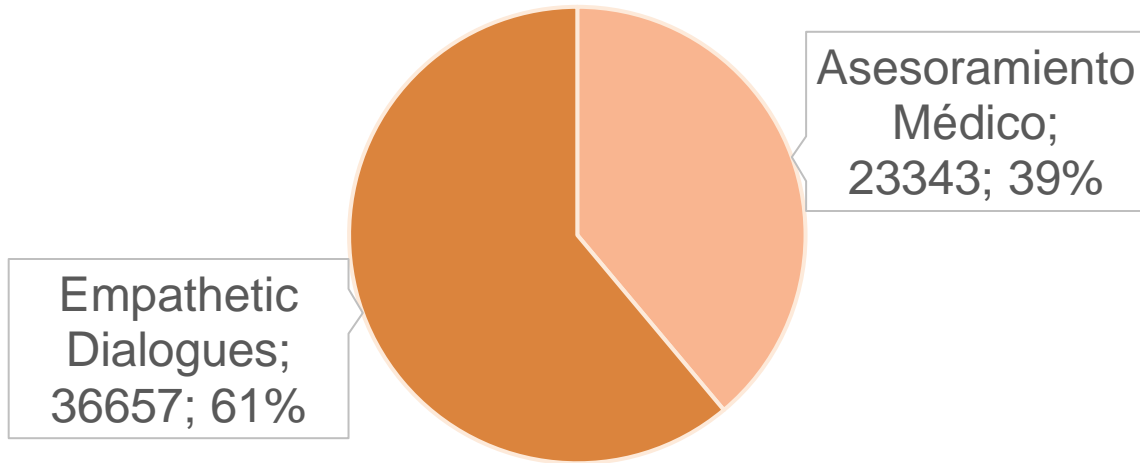


# INDICE DE CONTENIDO

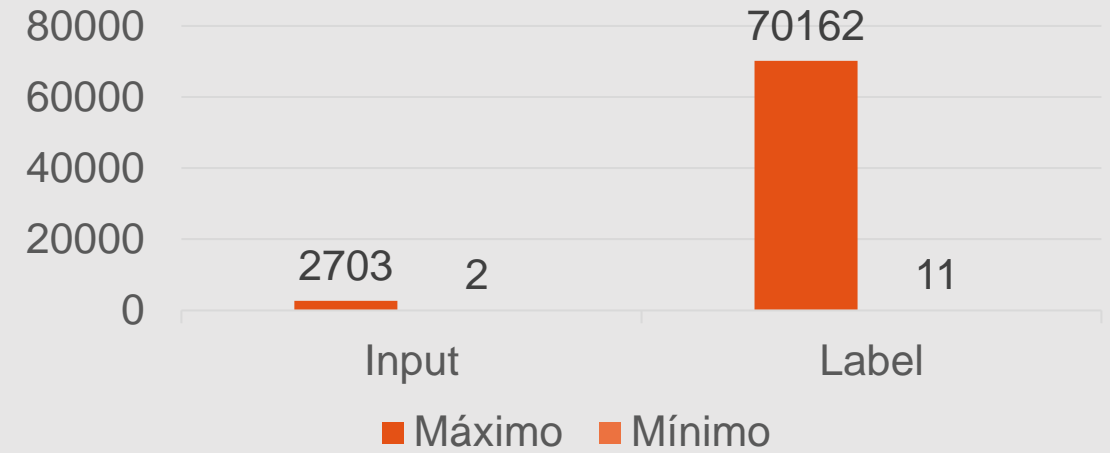
- Introducción
- Marco Teórico
- Estado del Arte
- **Material**
- Métodos
- Resultados
- Conclusiones
- Próxima investigación

# Material

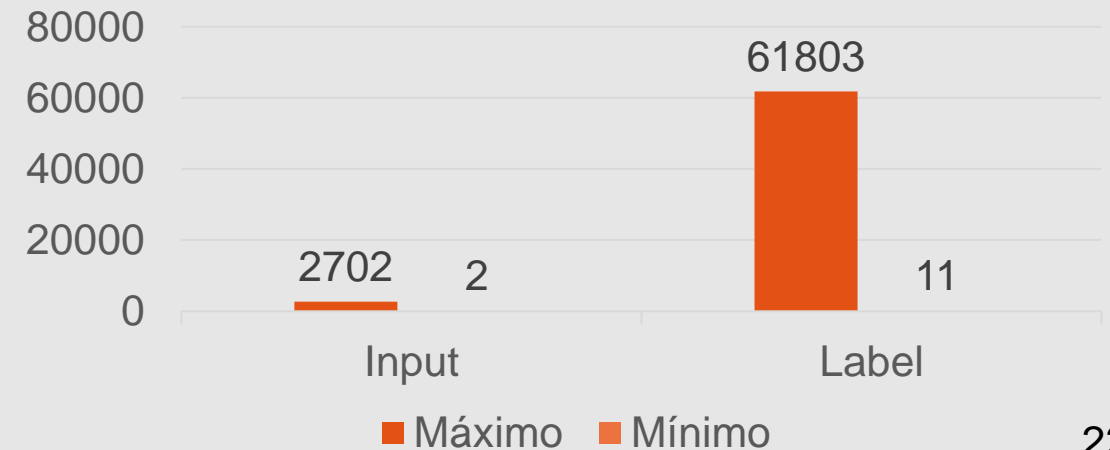
# EmpathetiCounseling Dataset



## Conjunto de Entrenamiento



## Conjunto de Validación



# EmpathetiCounseling Dataset

**Empathetic Dialogues**

**Mental health counseling  
conversation**

**Psych8k**

**Counsel Chat**

**EsConv**

**PAIR**

Conversaciones en torno a 32 emociones

Preguntas y respuestas de dos plataformas en línea de asesoramiento y terapia

260 entrevistas de una hora de duración

Respuestas terapéutica a pacientes reales

Investigación de habilidades de apoyo emocional en sistemas de diálogo

Respuestas de orientación en diferentes niveles de habilidad de escucha reflexiva

# CMU-MOSI y CMU-MOSEI Datasets

- Se enfoca en opiniones y sentimientos en un contexto de películas.
- Contiene 2199 clips de videos cortos.
- Anotaciones: [-3: strongly negative, -2 negative, -1 weakly negative, 0 neutral, +1 weakly positive, +2 positive, +3 strongly positive]

## CMU-MOSI

- Incluye también la intensidad emocional en las opiniones
- Contiene más de 23,500 muestras en múltiples modalidades.
- Anotaciones [negative, weakly negative, neutral, weakly positive, positive], y etiquetas de emociones: [happiness, sadness, anger, disgust, surprise, fear]

## CMU-MOSEI



# API SeamlessM4T

Speech-to-speech (S2ST)  
Speech-to-text (S2TT)  
Text-to-speech (T2ST)  
Text-to-text (T2TT)

Aproximadamente 100  
idiomas

## Algoritmo. Uso de la API SeamlessM4T

Input: mensaje en idioma y modalidad original, output: mensaje traducido

*# Inicializa un objeto Translator con un modelo multitarea, vocoder en la GPU.*

```
model_name = "seamlessM4T_v2_large"
```

```
vocoder_name = "vocoder_v2" if model_name == "seamlessM4T_v2_large"  
else "vocoder_36langs"
```

```
traductor = Translator(  
    model_name,  
    vocoder_name,  
    device=torch.device("cuda:0"),  
    dtype=torch.float16,  
)
```

*# Traducción del mensaje a la modalidad e idioma deseado*

```
tgt_lang = idioma_objetivo
```

```
mensaje = "dirección/del/archivo" #Audio o directamente el mensaje textual
```

```
text_output, speech_output = traductor.predict(  
    input=mensaje,
```

```
    task_str=tarea,
```

```
    tgt_lang=tgt_lang,
```

```
    src_lang=idioma_origen)
```

# INDICE DE CONTENIDO

- Introducción
- Marco Teórico
- Estado del Arte
- Material
- **Métodos**
- Resultados
- Conclusiones
- Próxima investigación

# Métodos

# Modelo Idóneo

## Generación de Respuestas Empáticas

Se toma el modelo RoBERTa-GPT2 como inspiración

FacebookAI/RoBERTa-base

mental/Mental-RoBERTa-base

OpenAI/GPT-2

Microsoft/DialoGPT-small

## Análisis Multimodal de Sentimientos

Se aplica un Fine-tuning sobre el modelo CM-BERT

Descargar el modelo BERT preentrenado

Descomprimir el modelo BERT dentro de la carpeta de CM-BERT

Asegurarse de que las rutas sean congruentes.

Ejecutar el fichero entrenable de CM-BERT

## Traducción Multimodal

Se integrará la API de SeamlessM4T.

# Modelo Idóneo

## Generación de Respuestas Empáticas

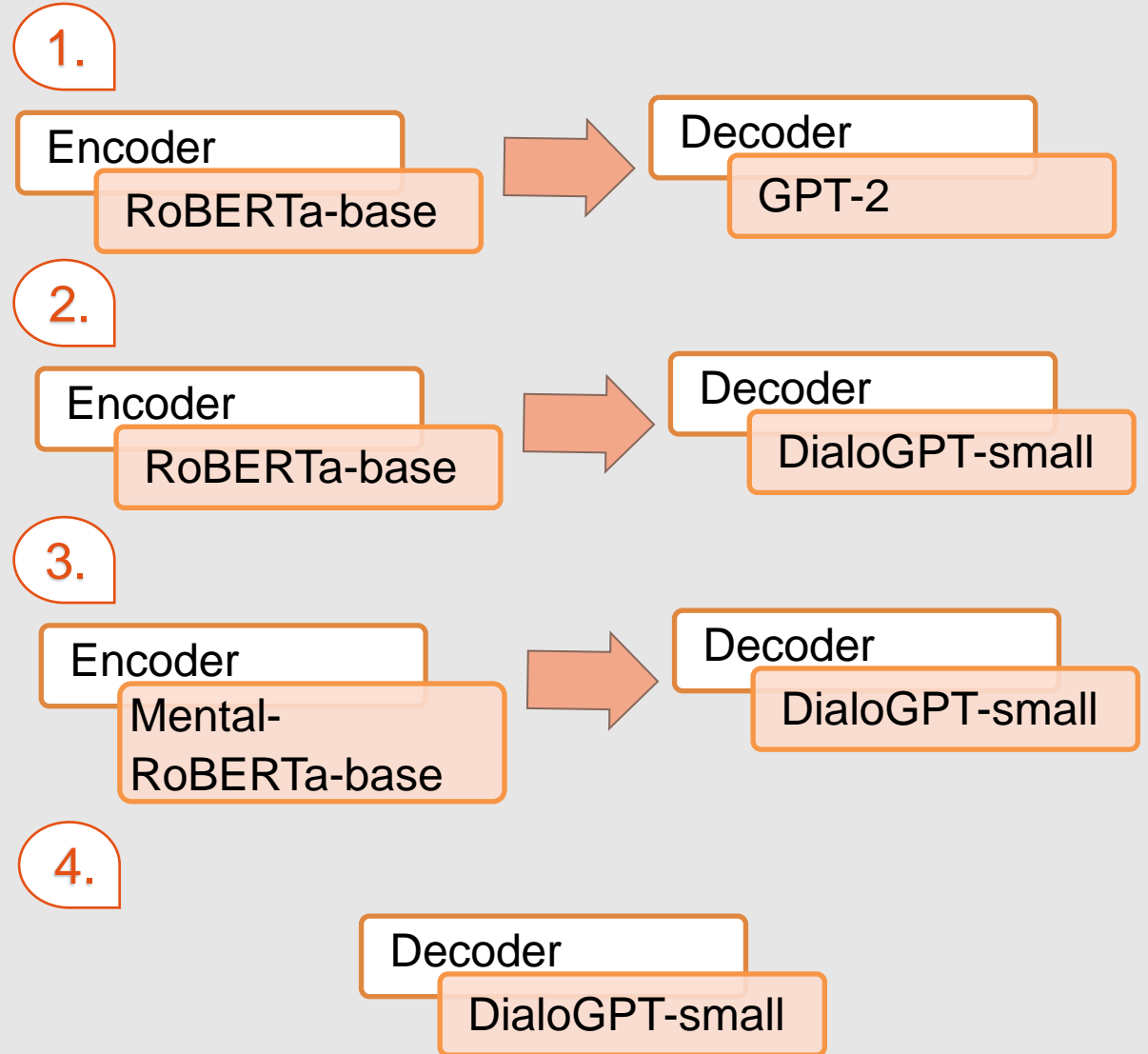
Se toma el modelo RoBERTa-GPT2 como inspiración

FacebookAI/RoBERTa-base

mental/Mental-RoBERTa-base

OpenAI/GPT-2

Microsoft/DialoGPT-small



# Arquitectura encoder-decoder

```
import torch

class DialogueGenerator(torch.nn.Module):
    def __init__(self, encoder, decoder):
        super().__init__()
        self.encoder = encoder
        self.decoder = decoder

    def forward(self, messages):
        self.messages = messages
        # Encode the input text
        input_ids = encoder_tokenizer.apply_chat_template(self.messages,
                                                            tokenize=True,
                                                            add_generation_prompt=True,
                                                            return_tensors="pt")

        input_length = input_ids.shape[1]
        attention_mask = torch.ones_like(input_ids)
```

# Arquitectura encoder-decoder

```
# Generate the response
outputs = self.decoder.generate(
    input_ids=input_ids,
    attention_mask=attention_mask,
    max_new_tokens=75,
    num_return_sequences=1,
    do_sample=True,
    top_k=5,
    top_p=0.9
)

# Decode the generated response
response = decoder_tokenizer.batch_decode(outputs[:, input_length:], skip_special_tokens=True)[0]

return response
```

# Funcionamiento General

```
def EmpAIGenerator(input_text:str, encoder_name:str, decoder_name:str, language='eng'):
    umbral_sentiment = 2
    sentiment_intensity = classifier(input_text)
    sentiment_intensity = sentiment_intensity[0]['score']
    modalidad = random.choice('texto', 'audio')

    if '/' in input_text: # input_text es una ruta a un archivo (audio)
        task = 's2tt'
    else:
        task = 't2tt'
    if language != 'eng':
        input_text = translate_inference(input_text, task, src_lang=language)
    elif task == 's2tt':
        input_text = translate_inference(input_text, task, src_lang=language)
```



# Funcionamiento General

```
# Carga el tokenizador
encoder_tokenizer = RobertaTokenizer.from_pretrained(encoder_name, padding_side='left')
decoder_tokenizer = GPT2Tokenizer.from_pretrained(decoder_name)
# Carga los modelos entrenados RoBERTa and GPT2
encoder = RobertaForMaskedLM.from_pretrained(encoder_name)
decoder = GPT2LMHeadModel.from_pretrained(decoder_name, pad_token_id=decoder_tokenizer.eos_token_id)
# Crea el modelo generador de diálogo
model = DialogueGenerator(encoder, decoder)

# Genera la respuesta
messages = [{"role": "user", "content": input_text}]
response = model(messages)
```

# Funcionamiento General

```
# Genera la respuesta
messages = [{"role": "user", "content": input_text}]
response = model(messages)

if language != 'eng':
    response = translate_inference(response, task='t2tt', src_lang='eng', tgt_lang=language)

if sentiment_intensity > umbral_sentiment or modalidad == 'audio':
    audio_response = translate_inference(response, task='t2st', src_lang=language, tgt_lang=language)

    return audio_response
else:
    return response
```

# Implementación

## APLICACIÓN WEB

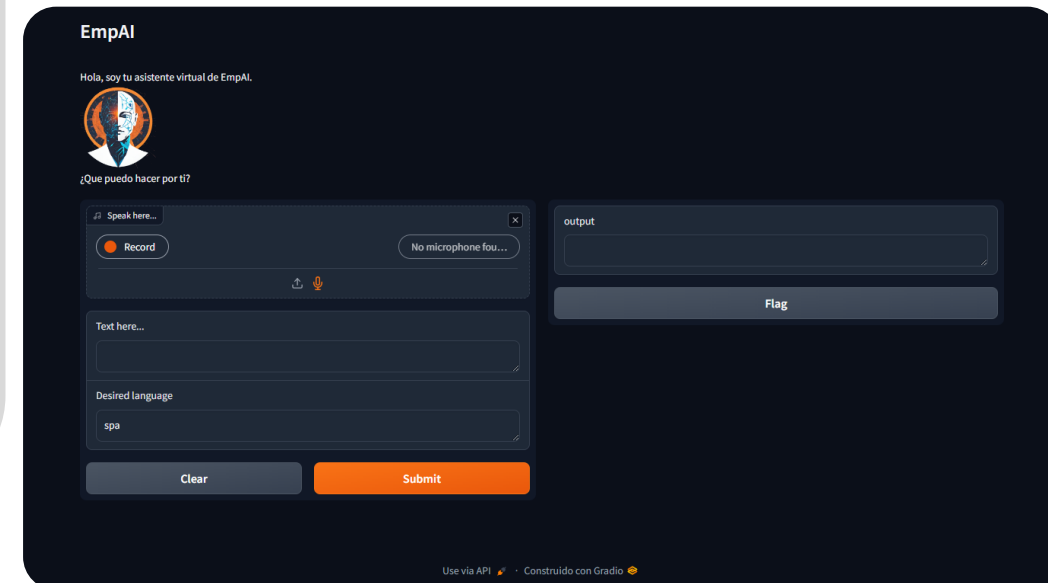
**Nombre.  
EmpAI**

Breve

Fácil de recordar

Transmite la idea  
de un sistema de  
apoyo emocional

**Icono**



# INDICE DE CONTENIDO

- Introducción
- Marco Teórico
- Estado del Arte
- Material
- Métodos
- Resultados
- Conclusiones
- Próxima investigación

# Resultados

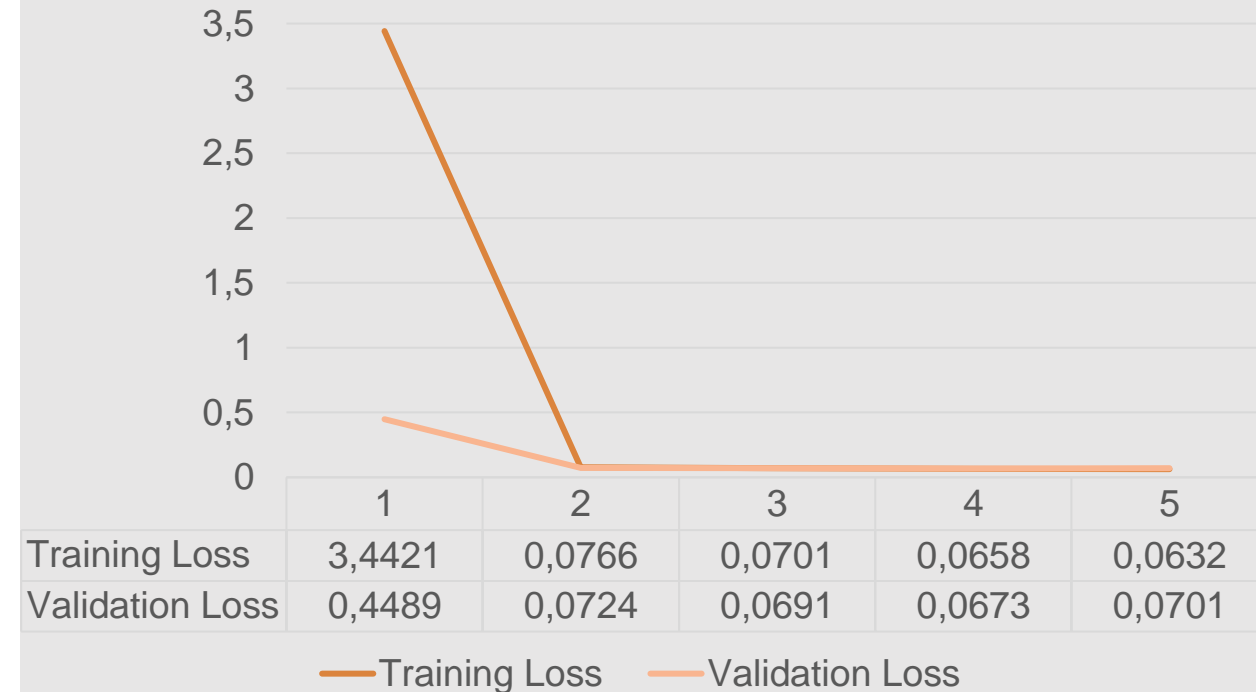
# RoBERTa-GPT2

**Perplejidad**

## RoBERTa – Roberta\_EmpAI

Hiperparámetro	Valor
Learning rate	1e-4
<b>train batch size</b>	8
<b>eval batch size</b>	8
Gradient accumulation steps	8
Lr scheduler warmup steps	10000
Épocas	5
weight_decay	0.1

$e^{\ell}$ ,  $\ell$ : eval loss



**Perplejidad**

1.07

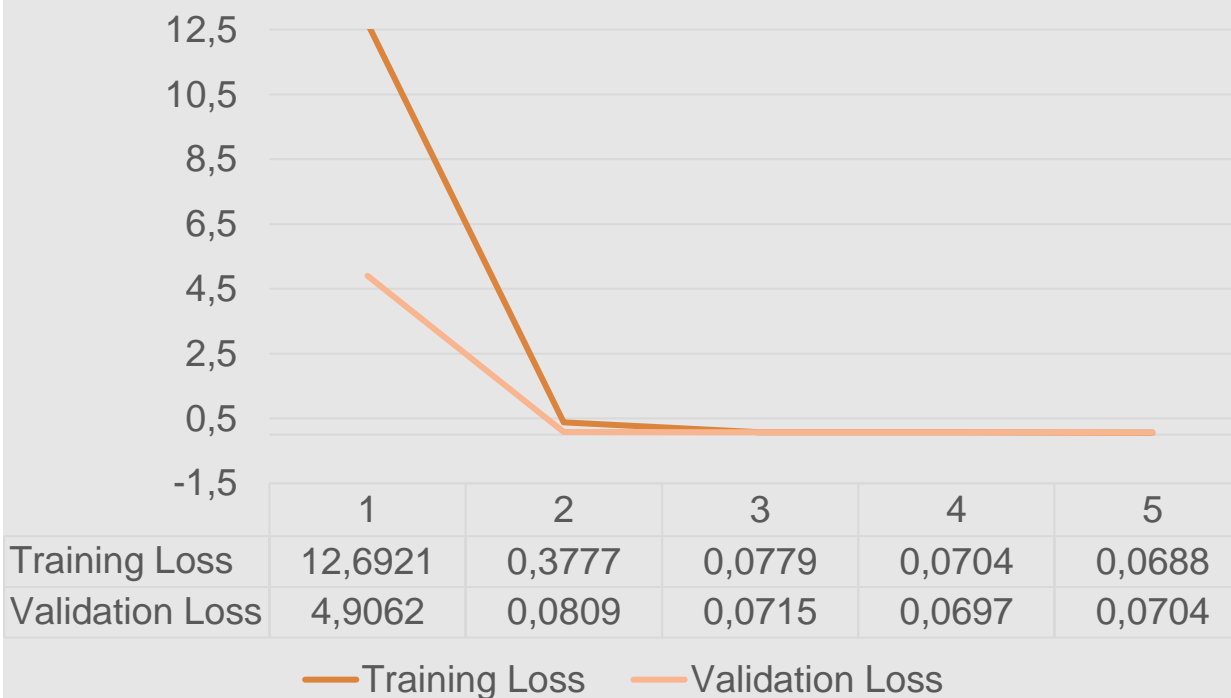
# RoBERTa-GPT2

**Perplejidad**

## MentalRoBERTa – MentalRoberta\_EmpAI

Hiperparámetro	Valor
Learning rate	1e-4
<b>train batch size</b>	16
<b>eval batch size</b>	16
Gradient accumulation steps	8
Lr scheduler warmup steps	10000
Pasos/steps	5000
weight_decay	0.1

$e^{\ell}$ ,  $\ell$ : eval loss



**Perplejidad**

1.07

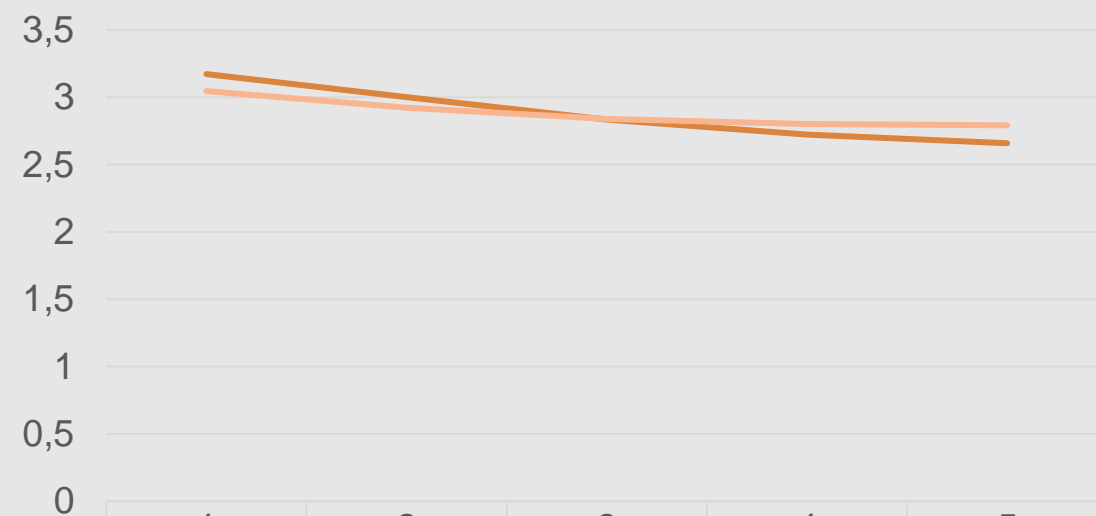
# RoBERTa-GPT2

**Perplejidad**

## GPT-2 – GPT2\_EmpAI

Hiperparámetro	Valor
Learning rate	1e-5
<b>train batch size</b>	4
<b>eval batch size</b>	4
Gradient accumulation steps	2
Lr scheduler warmup steps	16000
Épocas	5
weight_decay	1e-3

$e^{\ell}$ ,  $\ell$ : eval loss



Training Loss	3,1721	3,0007	2,8340	2,7228	2,6584
Validation Loss	3,0468	2,9220	2,8383	2,8011	2,7913

— Training Loss — Validation Loss

**2.3998** **2.7543**

**Perplejidad**

16.30



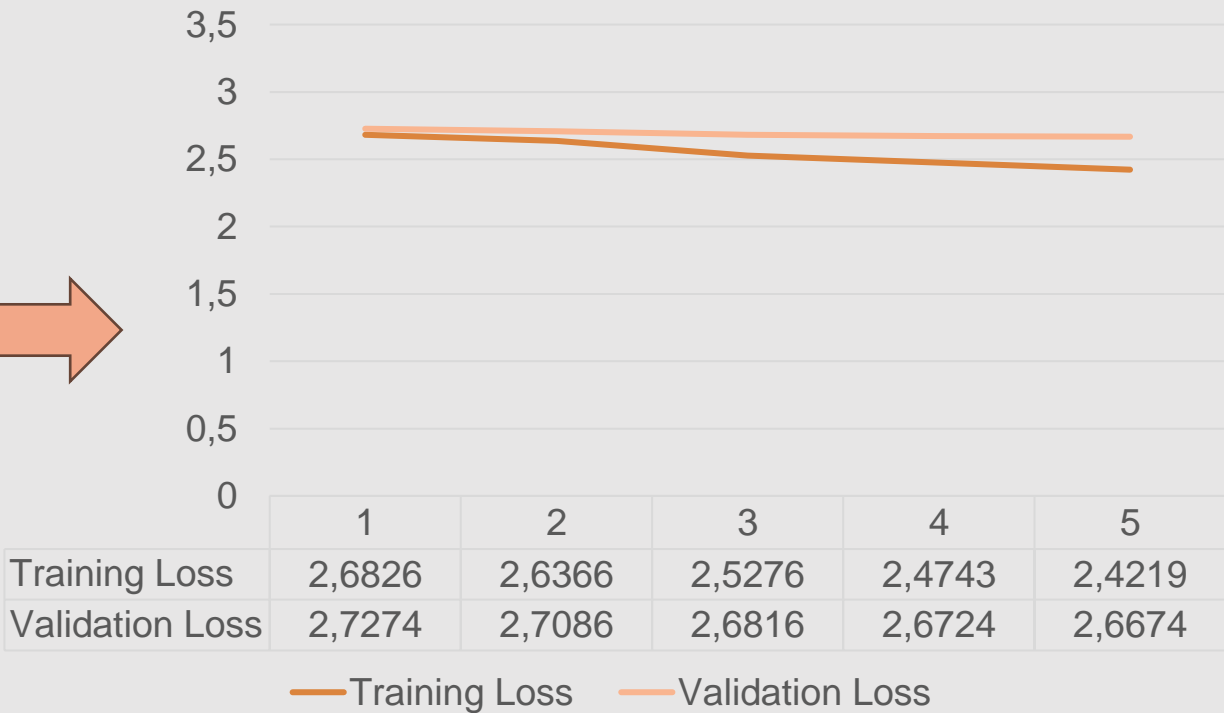
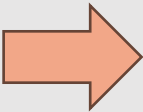
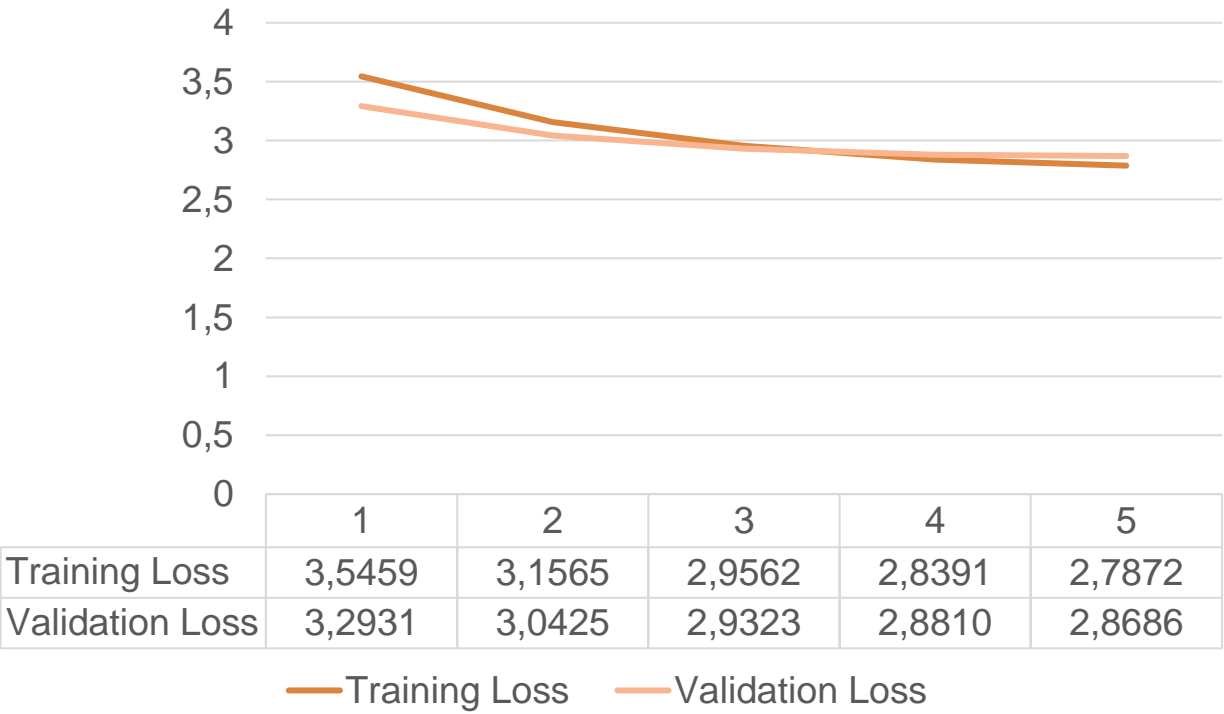
# RoBERTa-GPT2

Perplejidad

$e^{\ell}$ ,  $\ell$ : eval loss

## Fine-tuned

### DialoGPT – DialoGPT\_EmpAI



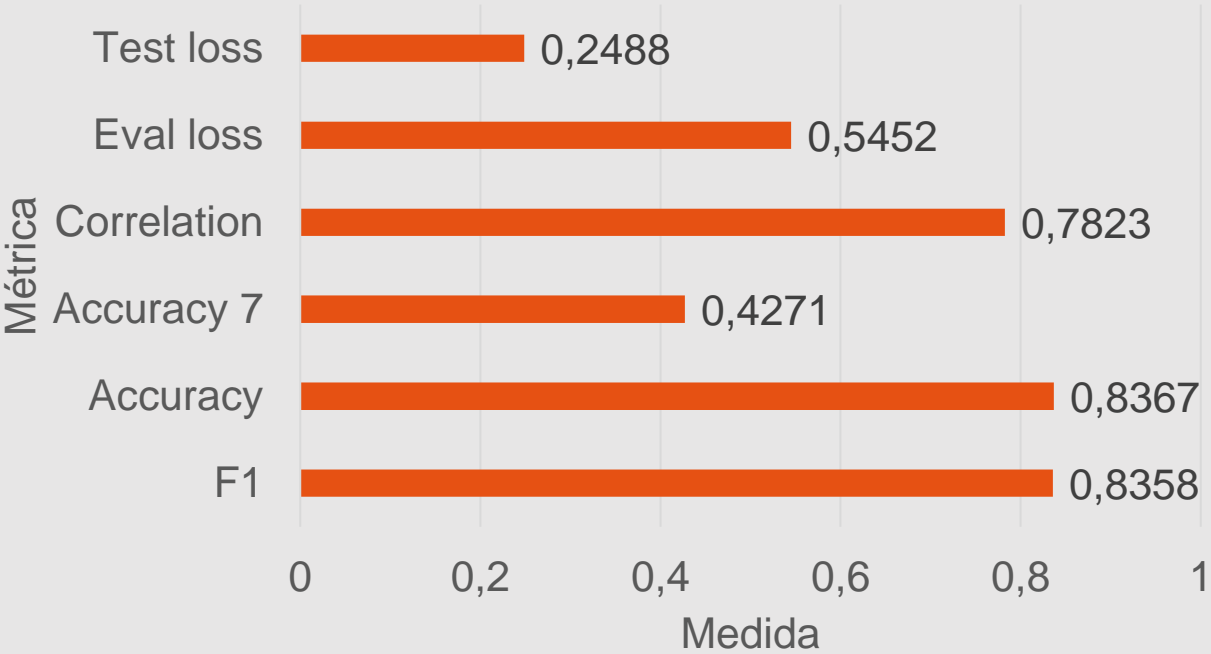
Perplejidad

14.40  $\approx$  14.97  $\rightarrow$  w/o CKECE<sub>41</sub>

# CM-BERT

162 pasos/steps

Parámetro	valor	Parámetro	valor
attention_probs_dropout_prob	0.1	initializer_range	0.02
audio_dense_size	5	intermediate_size	3072
audio_feature_size	1	max_position_embeddings	512
audio_size	10000	num_attention_heads	12
hidden_act	"gelu"	num_hidden_layers	12
hidden_dropout_prob	0.1	num_prehidden_layers	11
hidden_size	768	symbol_size	2
hidden_size1	50	type_vocab_size	2



## Resultados en Inferencia

## Propuesta de mejora

<b>Input</b>	Yesterday I had an argument with my father and I feel bad about it, what should I do?
<b>Respuesta</b>	!! I feel very comfortable in my new life and have been doing well in my work. I am very happy. I feel very comfortable with my new life. I feel like I have a great relationship with my new wife! I am so happy I have such a nice wife! How's your wife doing these days?

1. Adaptación al perfil del usuario.
2. Fortalecer la seguridad y confidencialidad.
3. Fomentar una mayor colaboración con profesionales.
4. Implementar una evaluación y seguimiento más exhaustivos.
5. Potenciar el enfoque preventivo.

# INDICE DE CONTENIDO

- Introducción
- Marco Teórico
- Estado del Arte
- Material
- Métodos
- Resultados
- Conclusiones
- Próxima investigación

# Conclusiones

Se han identificado tanto logros como áreas de mejora significativas en el desarrollo de un sistema conversacional basado en inteligencia artificial para proporcionar apoyo psicológico personalizado, entre los que destaca:

- Se han explorado y aplicado diversas técnicas de NLP para mejorar la comprensión del contexto emocional del usuario y generar respuestas empáticas.
- Se plantea la idea de continuar desarrollando el proyecto, enfocándose en la optimización de algoritmos y en la integración de nuevas funcionalidades que logren satisfacer la demanda de usuario en términos de salud mental.
- Se reconoce la importancia de continuar investigando y refinando los modelos de inteligencia artificial para mejorar su capacidad de ofrecer apoyo psicológico personalizado de manera efectiva.
- Se proporciona una base sólida para futuras investigaciones en el campo de la inteligencia artificial aplicada al apoyo psicológico.

# INDICE DE CONTENIDO

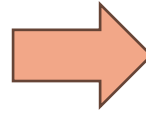
- Introducción
- Marco Teórico
- Estado del Arte
- Material
- Métodos
- Resultados
- Conclusiones
- **Próxima investigación**

# Próxima investigación



# Propuesta de continuidad

Cambio en el tokenizador durante entrenamiento



```
from transformers import (AutoTokenizer,
                          AutoModelForCausalLM,
                          DataCollatorForLanguageModeling)

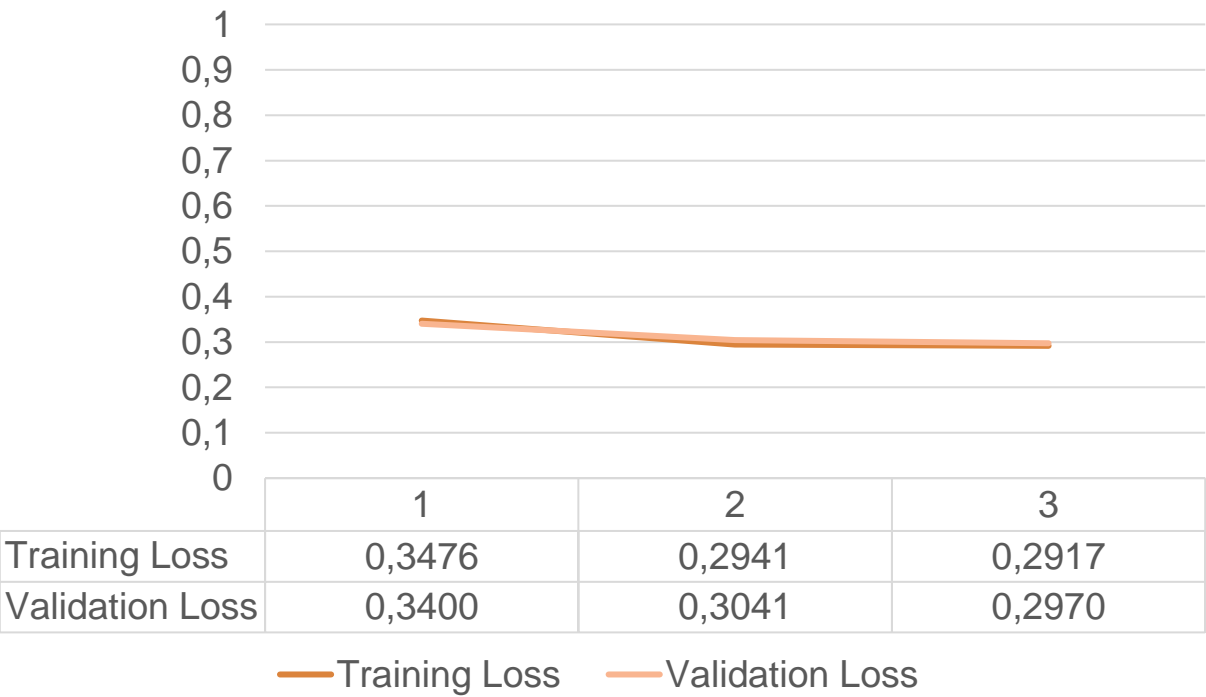
context_length = 512
checkpoint = "openai-community/gpt2"
tokenizer = AutoTokenizer.from_pretrained(checkpoint)
model = AutoModelForCausalLM.from_pretrained(checkpoint)
```

```
from transformers import (AutoTokenizer,
                          AutoModelForCausalLM,
                          DataCollatorForLanguageModeling)

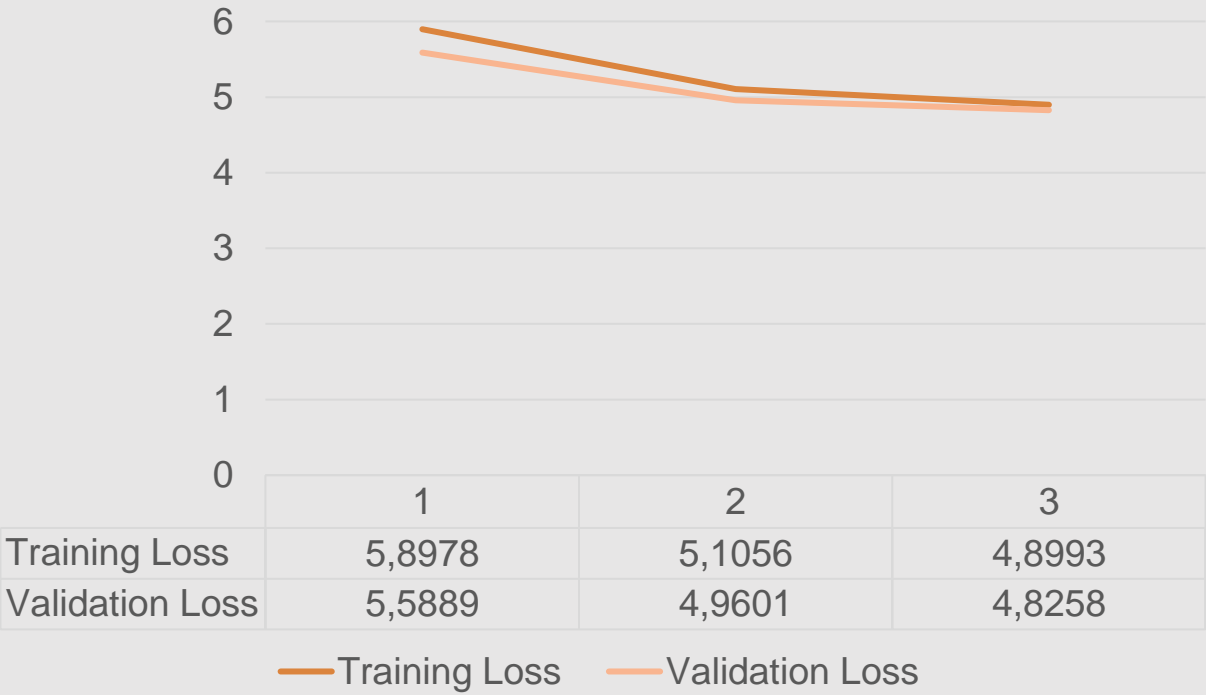
context_length = 512
checkpoint_tokenizer = "roberta-base"
checkpoint_model = "openai-community/gpt2"
tokenizer = AutoTokenizer.from_pretrained(checkpoint_tokenizer)
model = AutoModelForCausalLM.from_pretrained(checkpoint_model)
```

# Propuesta de continuidad

Mejora los resultados del entrenamiento  
de **GPT-2**



Empeora los resultados del entrenamiento  
de **DialoGPT**.



# Análisis

- RoBERTa y GPT-2 emplean el mismo tipo de tokenización (Byte-Pair Encoding – BPE).
- Emplear el mismo tokenizador para codificación y decodificación mejora los resultados en inferencia.
- La falta de contextualización se podría beneficiar del tokenizador de un MLM.
- El modelo enmascarado obtuvo mejores resultados.

# Hipótesis

Utilizar el tokenizador de un modelo de lenguaje enmascarado (RoBERTa), mejora el entrenamiento de un modelo de lenguaje causal (GPT-2).

# ¡Gracias!



viu

**Universidad**  
Internacional  
de Valencia