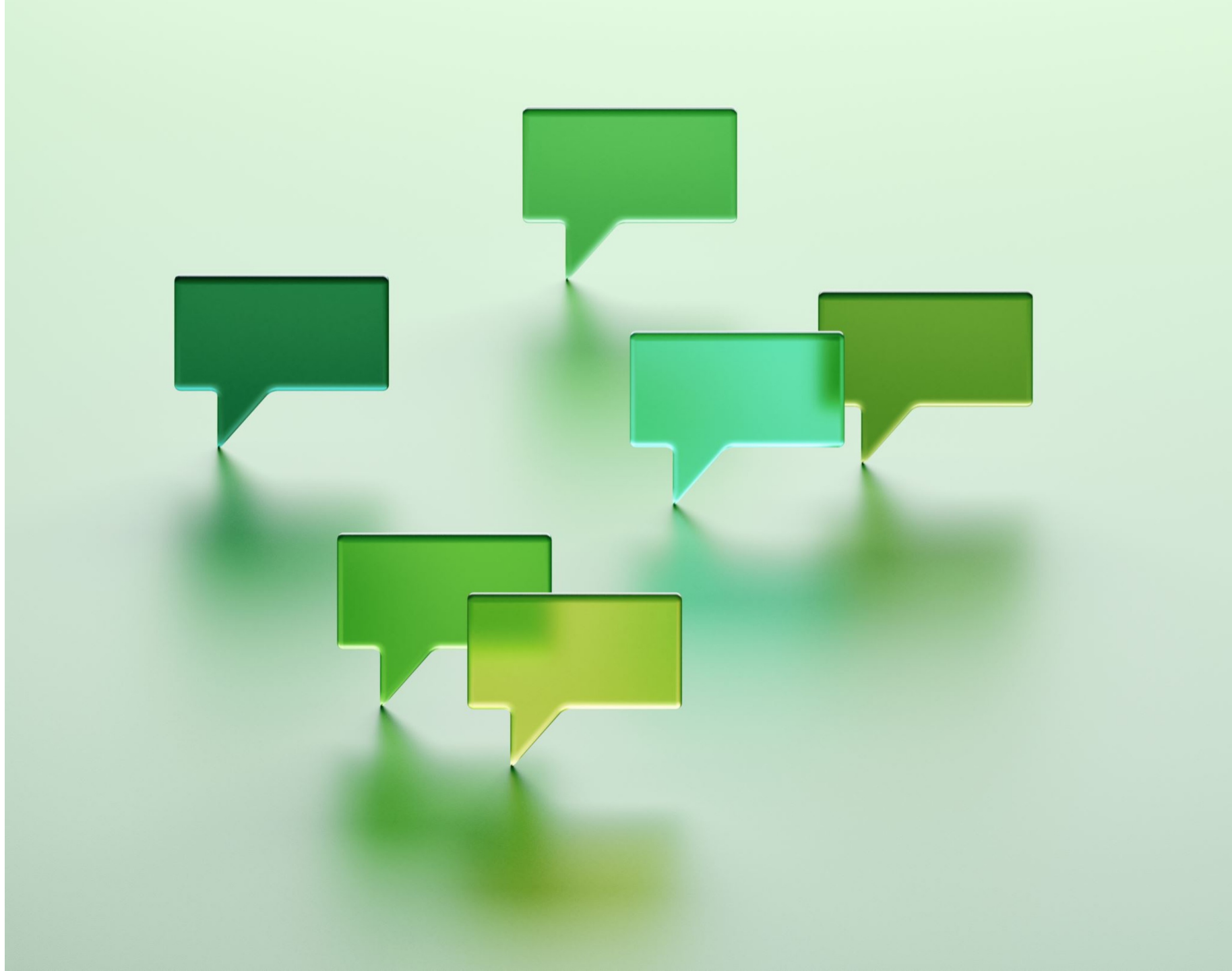
Modelado del lenguaje con Transformers

Dra. Helena Gómez Adorno, IIMAS, UNAM
helena.gomez@iimas.unam.mx

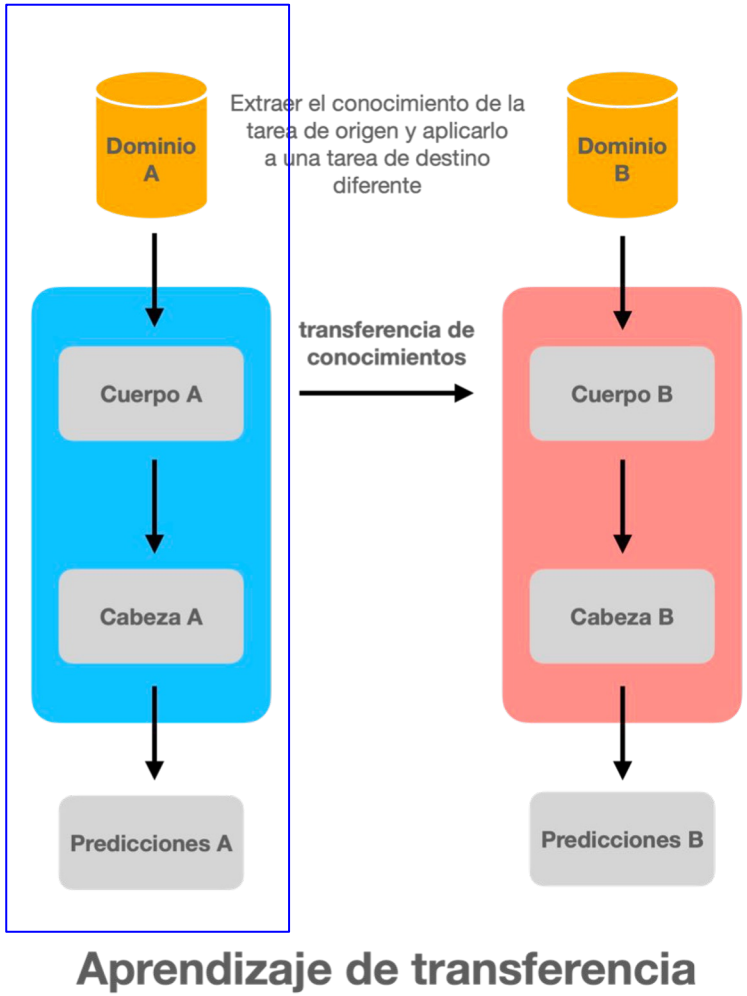
Miguel Angel Alvarez, CIMAT
miguel.alvarez@cimat.mx

Victor Giovanni Morales Murillo, LKE, FCC, BUAP
vg0552010@gmail.com



Modelado del lenguaje

- El modelado de lenguaje predice palabras en un enunciado.
- Utiliza el aprendizaje autosupervisado, es decir, no requiere de etiquetas.
- Existen dos principales estilos de modelado del lenguaje: autorregresivo y enmascarado.

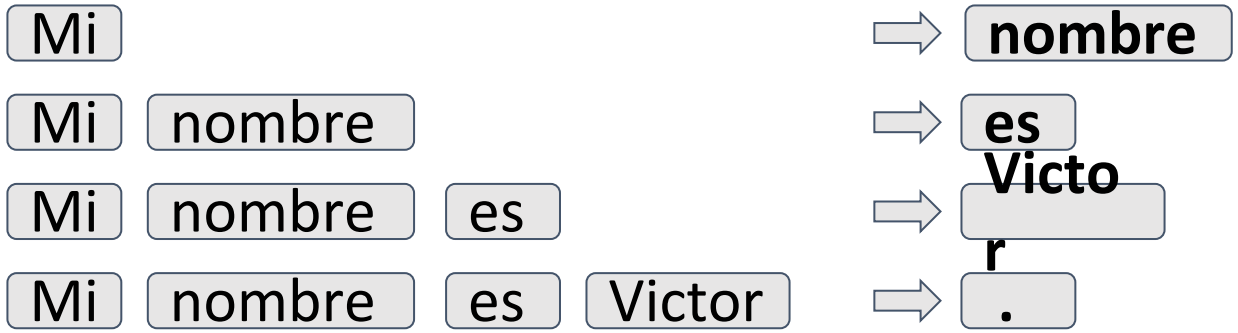


Aprendizaje de transferencia

https://huggingface.co/docs/transformers/main/es/tasks/language_modeling

Modelado autorregresivo

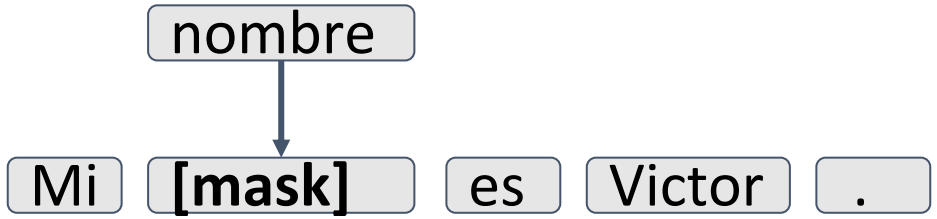
El modelado de lenguaje autorregresivo o causal predice el siguiente token en una secuencia de tokens y el modelo solo puede considerar los tokens a la izquierda.



https://somosnlp.org/nlp-de-cero-a-cien/sesion-04

Modelado enmascarado

El modelado de lenguaje por enmascaramiento predice un token enmascarado en una secuencia y el modelo puede considerar los tokens bidireccionalmente.



https://somosnlp.org/nlp-de-cero-a-cien/sesion-04

Modelos basados en BERT

1.1 RoBERTa

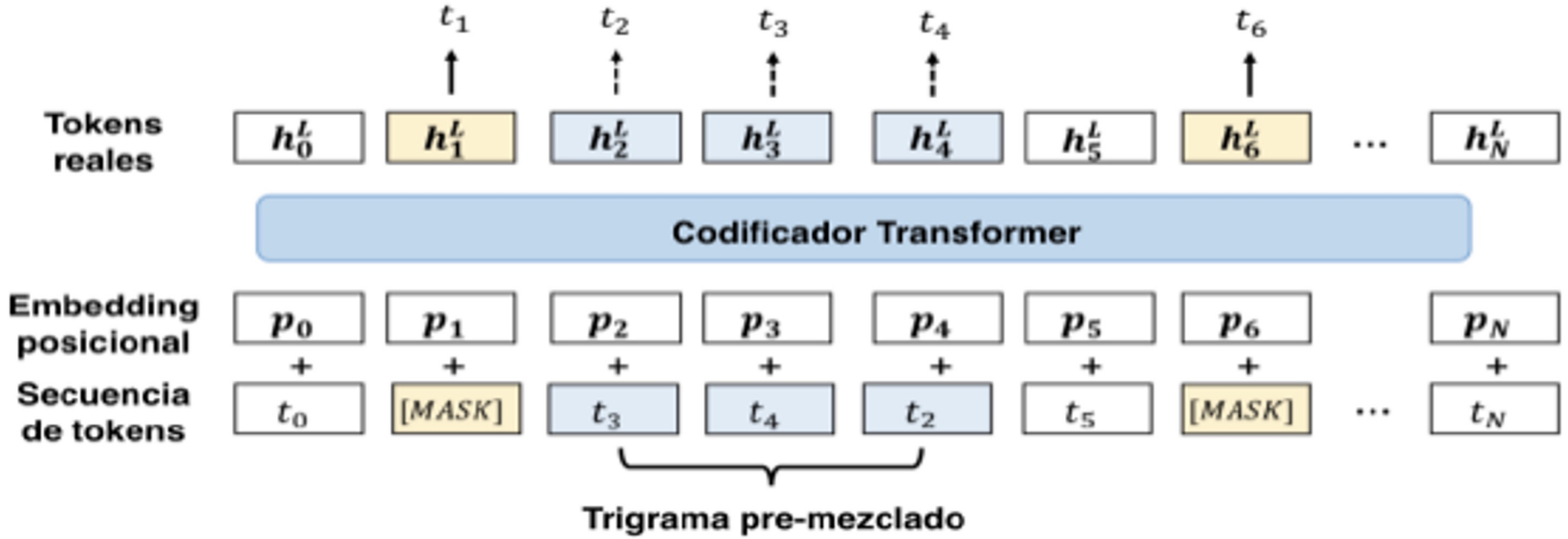
- Modificación de la etapa de pre-entrenamiento.
- Fue propuesto como una configuración optimizada de BERT.
- Utiliza la misma arquitectura y las mismas variantes de tamaño con 12 y 24 bloques Transformer.
- La tarea de modelado de lenguaje se aborda con un enmascaramiento dinámico.
- Elimina la tarea de predicción de próxima oración.
- Utiliza lotes más grandes y secuencias de texto más largas.
- Usa un conjunto de datos más extenso formado por los conjuntos de datos de BookCorpus, CC-News, Open WebText y Stories.

1.2 StructBERT

- Modificación de la etapa de pre-entrenamiento.
- Fue propuesto como una extensión de BERT.
- Incorpora información estructural del lenguaje al pre-entrenamiento.
- La tarea de modelado de lenguaje se aborda con el ordenamiento a nivel de palabras y a nivel de enunciados.
- Conjunto de datos BookCorpus y Wikipedia en inglés.

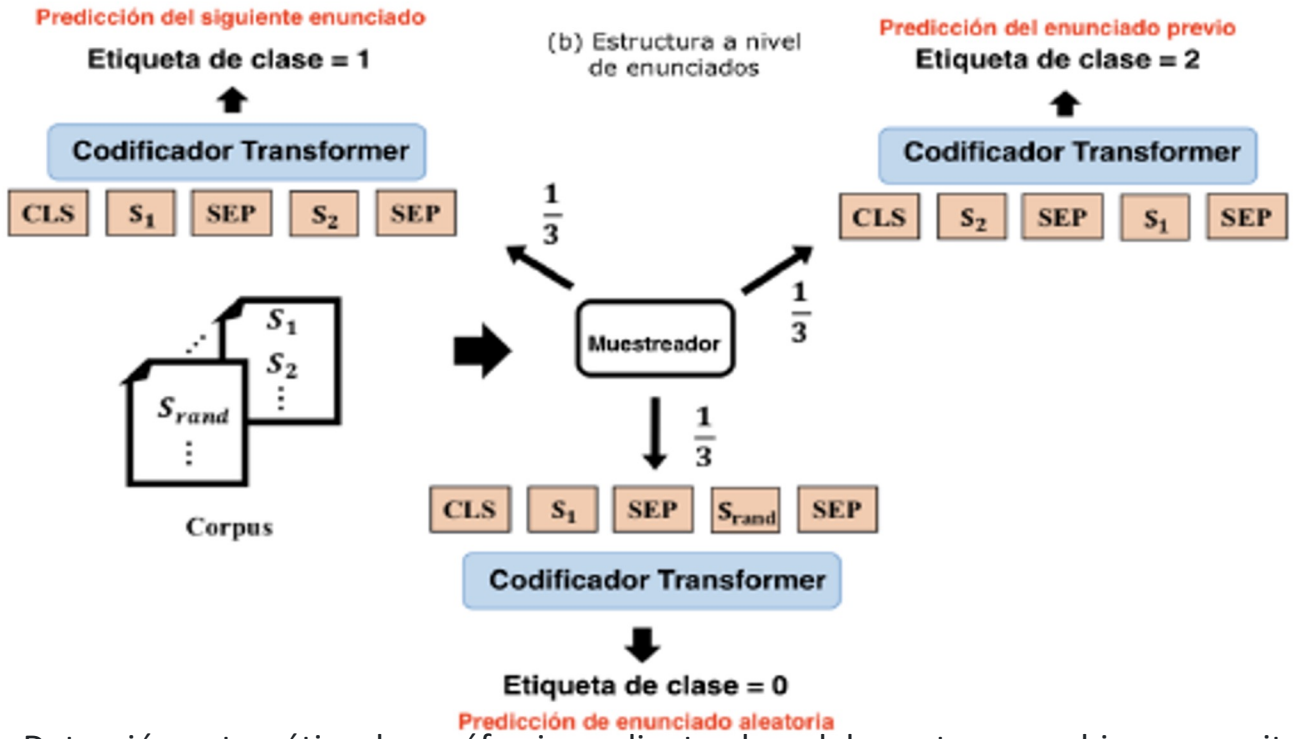
Ordenamiento a nivel de palabras

- Enmascaramiento del 15 % de los tokens de la secuencia de entrada.
- Se selecciona un cierto porcentaje de los tokens y se cambia el orden de forma aleatoria.
- Se obtienen los vectores de salida de los bloques Transformer y se agregan a un clasificador Softmax para predecir el orden correcto de los tokens.



Ordenamiento a nivel de enunciados

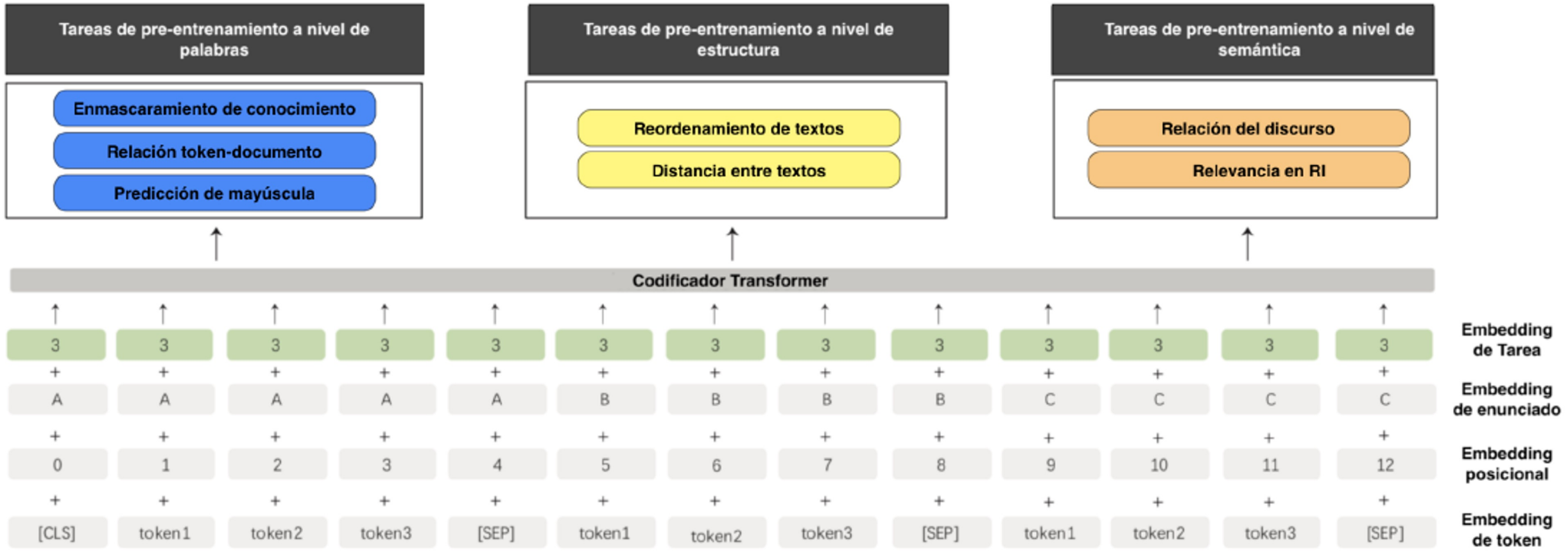
- Dado un par de texto S1 y S2, el modelo se entrena para aprender a predecir:
  - si S2 es el texto que sigue de S1.
  - si S2 es el texto que precede a S1.
  - Si corresponde a otro texto aleatorio de otro documento.



1.3 Ernie 2.0

- Modificación de la etapa de pre-entrenamiento.
- Marco de trabajo multi-tarea continuo para aprender información léxica, sintáctica y semántica.
- Utiliza el conocimiento adquirido en las tareas previas para adaptarlo a las nuevas tareas de pre-entrenamiento.
- Comparte un bloque codificador en cada una de las tareas.
- Utilizan 7 tareas de pre-entrenamiento divididas en tres conjuntos.

1.3 Ernie 2.0


Tareas de pre-entrenamiento a nivel de palabras
Enmascaramiento de conocimiento
Relación token-documento
Predicción de mayúscula
Tareas de pre-entrenamiento a nivel de estructura
Reordenamiento de textos
Distancia entre textos
Tareas de pre-entrenamiento a nivel de semántica
Relación del discurso
Relevancia en RI
Codificador Transformer
3 3 3 3 3 3 3 3 3 3 3 3 3
A A A A A B B B B C C C C
0 1 2 3 4 5 6 7 8 9 10 11 12
[CLS] token1 token2 token3 [SEP] token1 token2 token3 [SEP] token1 token2 token3 [SEP]
Embedding de Tarea
Embedding de enunciado
Embedding posicional
Embedding de token

Tareas a nivel de palabras

- Enmascaramiento de conocimiento: Una fracción de las entidades nombradas y frases del texto son remplazadas por el token [MASK] y el modelo aprende a predecir el valor real de dichos tokens.

- Relación token-documento: Aprende a predecir si el token de un segmento aparece en otros segmentos del documento original. Identifica la relevancia de las palabras en un documento.

- Predicción de mayúsculas: Identifica si una palabra inicia con letra mayúscula o no.

Tareas a nivel de estructura

- Reordenamiento del texto: Dado un párrafo se divide en m segmentos y son mezclados aleatoriamente y se debe predecir la k-clase de cada segmento.

- Distancia entre textos: Dado dos textos se debe determinar si son textos adyacentes en el mismo documento, si están en el mismo documento pero no son adyacentes y si no pertenecen al mismo documento.

Tareas a nivel semántico

- Relación del discurso: El modelo aprende a predecir la relación semántica de dos textos.

- Relevancia en un sistema de recuperación de información: Dada una consulta compuesta de un primer texto y dado un título compuesto por un segundo texto, el sistema identifica si existe una relevancia fuerte, débil o nula.

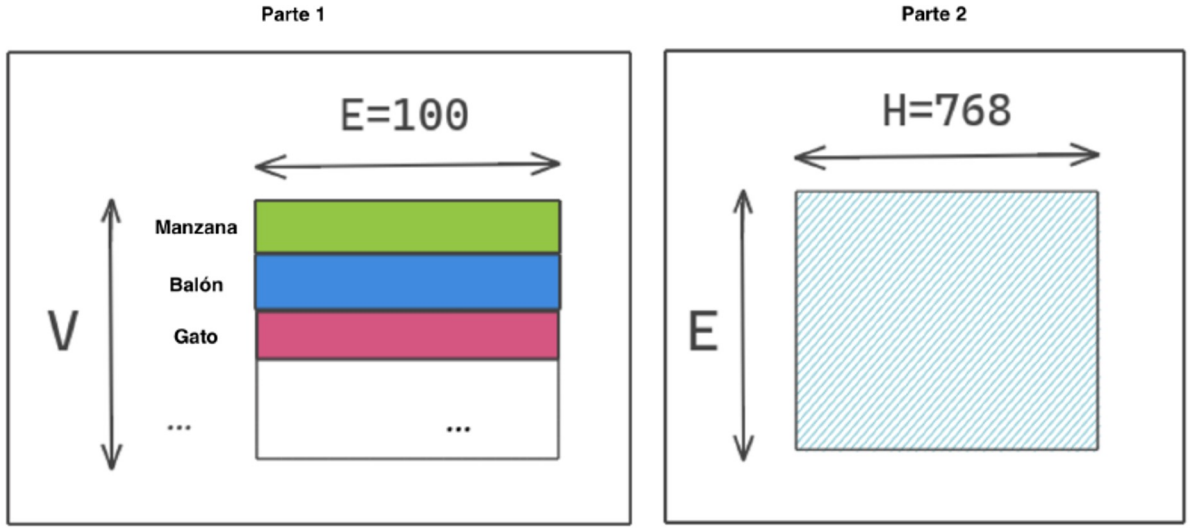
2.1 ALBERT

- Reducción de tamaño del modelo del lenguaje.
- Se utiliza en entornos con memorias GPUs y TPUs limitadas.
- 2 técnicas son propuestas para reducir el modelo BERT.

- Compartición de parámetros: En lugar de aprender pesos únicos para cada bloque Transformer, se aprende únicamente pesos para un bloque y este se reutiliza las veces que sea necesario. Se reducen de 110 millones de parámetros la versión base a únicamente 31 millones de parámetros utilizando los mismos bloques.

2.1 ALBERT

- Parametrización de embeddings factorizada:
- Al descomponer la matriz de embeddings del vocabulario en dos matrices más pequeñas permite separar el tamaño de las capas ocultas del modelo del tamaño del vocabulario. Facilita el crecimiento del tamaño de las representaciones ocultas sin afectar significativamente el crecimiento del vocabulario.
- Proyecta los vectores de entrada en un espacio vectorial E de menor dimensión, en este caso E = 100 y posteriormente proyectarlos sobre el espacio vectorial de las representaciones ocultas, donde E=H=768 en la versión base.

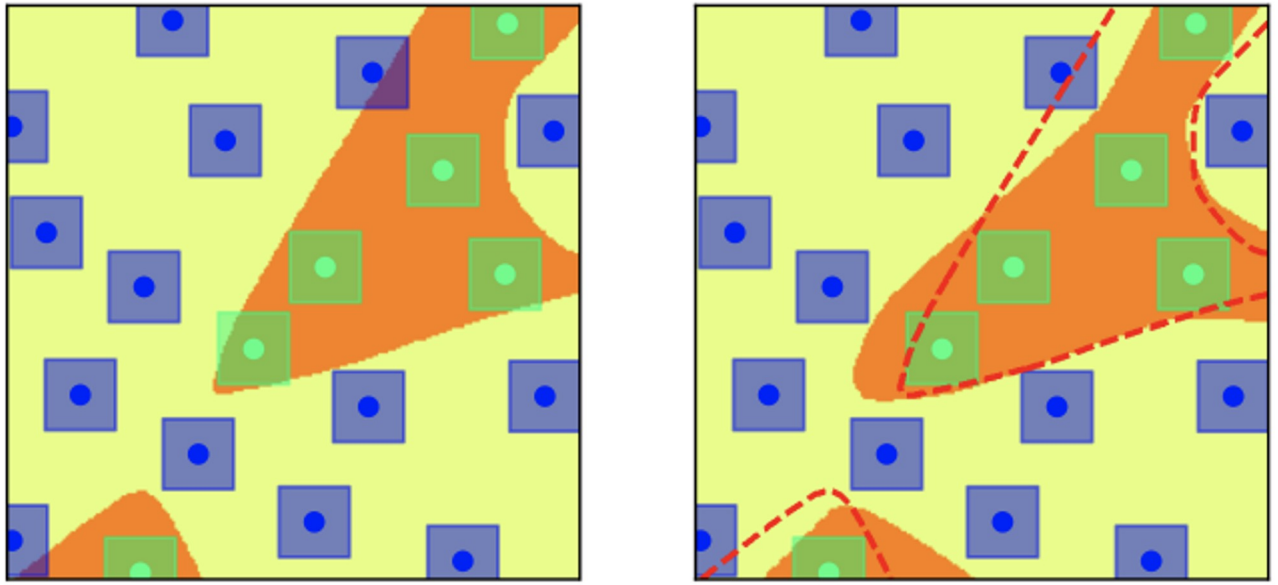


3.1 Algoritmo SMART

- Modificación al algoritmo de ajuste fino.
- Debido a la limitada cantidad de datos en las tareas objetivo y la complejidad de las arquitecturas en el estado del arte, los modelos del lenguaje son propensos a sobre ajustarse.
- Regularización adversaria inductora de suavidad: Se agrega una leve perturbación a los ejemplos de entrada para controlar la capacidad del modelo y su alta complejidad.

3.1 Algoritmo SMART

- En la imagen, se observan los límites de decisión aprendidos sin regularización (a) y con regularización (b).



3.1 Algoritmo SMART

- Método de optimización:
- Basado en los métodos de Bregman, cuya idea es imponer una penalización en cada iteración para evitar la actualización agresiva de los parámetros del modelo.

4.1 DeBERTA

- Modificación a la arquitectura Transformer.
- Se incorpora la atención desenredada:
- Las palabras son codificadas usando 2 vectores, uno en el que se captura el contenido o significado y otro en el que se codifica la posición relativa dentro del texto.
- Resuelve la limitante de BERT en el que los vectores de contenido y posición se suman para obtener la representación vectorial de un token.
- Permitiendo que una misma palabra tenga vectores distintos para posiciones diferentes.