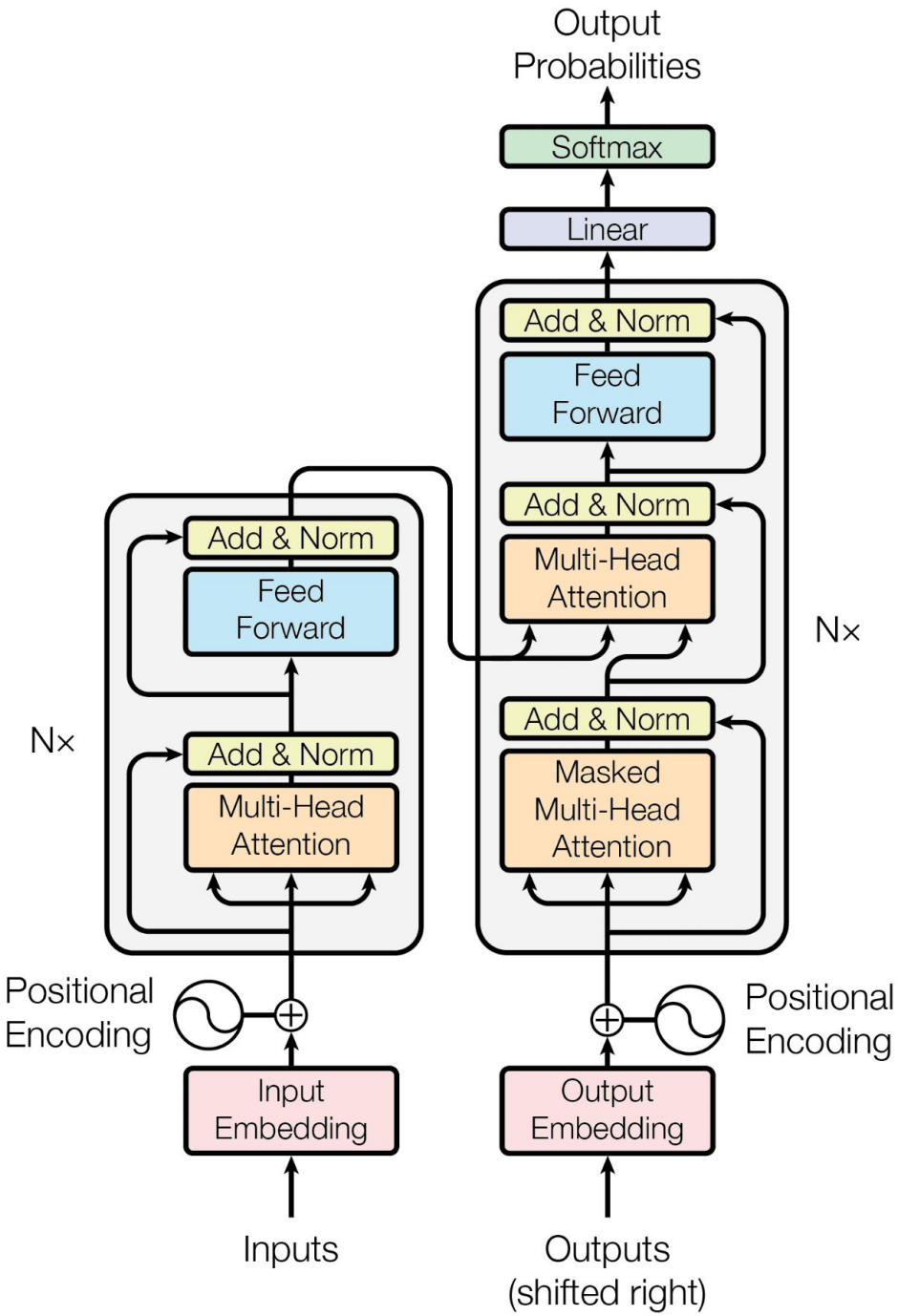
Redes Neuronales para Lenguaje Natural

2024

Grupo de Procesamiento de Lenguaje Natural
Instituto de Computación


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

Formas de uso

El transformer original fue planteado como arquitectura encoder-decoder

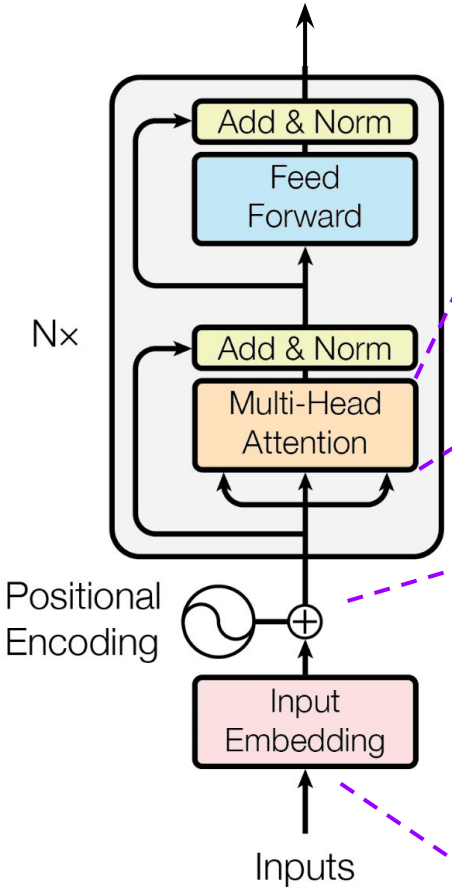
Primer ejemplo de uso: traducción automática

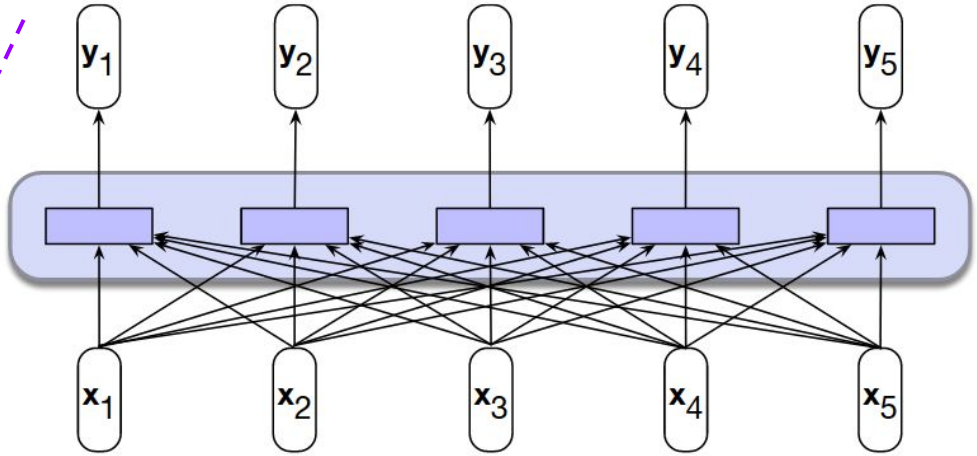
Pero luego aparecen otras variantes

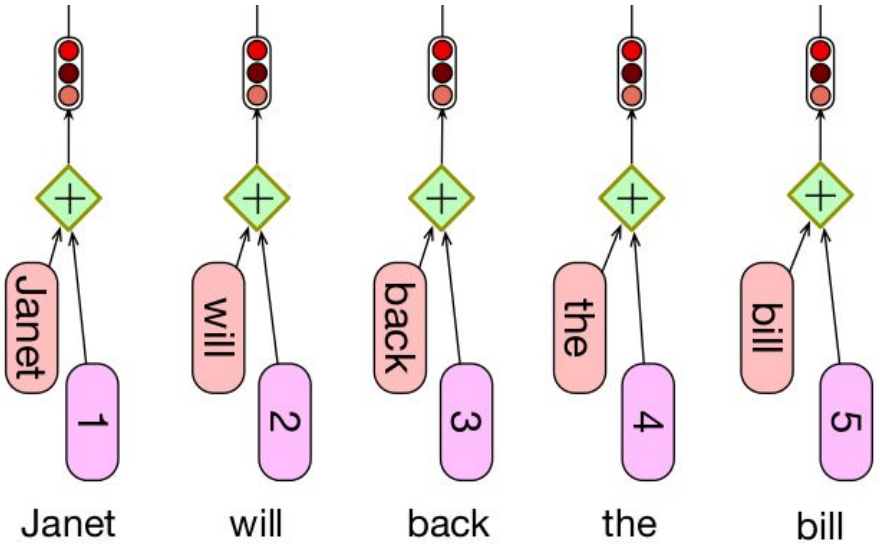
Encoder-only (por ejemplo BERT)

Decoder-only (por ejemplo GPT)

…
Varias capas, pero siempre del mismo tamaño de salida
…







BERT: Bidirectional Transformers

BERT

Bidirectional Encoder Representation from Transformers

Es un modelo que nos permite construir embeddings contextuales

- Embeddings para cada token en su contexto
- Embedding total para la secuencia entera de tokens

BERT no es la única técnica para hacer embeddings contextuales

También está ELMO: Embeddings from Language Models

BERT

Utiliza el transformer para crear un modelo de lenguaje

Pero no está hecho para predecir la siguiente palabra

Dos objetivos de entrenamiento:

- Masked Language Model: Dada una secuencia a la que le faltan algunas palabras (en cualquier lado), predecirlas
- Next Sentence Prediction: Dadas dos secuencias, predecir si una es continuación natural de la otra en un texto

Para esto agrega nuevos tokens al vocabulario: [MASK], [CLS] y [SEP]

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Devlin, Chang, Lee, Toutanova, (2018)

Masked Language Model

MLM es la tarea de predecir palabras “enmascaradas” en una secuencia

Es un ejemplo de tarea de eliminación de ruido o *denoising*

Dada una secuencia con palabras faltantes, intentar predecirlas (en inglés es *fill-in-the-blanks* o *cloze test*)

- El puma _____ sus garras en la desafortunada presa .
- El cielo _____ cubierto de _____ que parecían de algodón .

Masked Language Model

BERT original propone tomar cada secuencia (oración) de entrenamiento y perturbar 15% de los tokens de la siguiente manera:

- 80% se sustituye por [MASK]
- 10% se sustituye por otro token aleatoriamente
- 10% restante se mantiene sin cambios

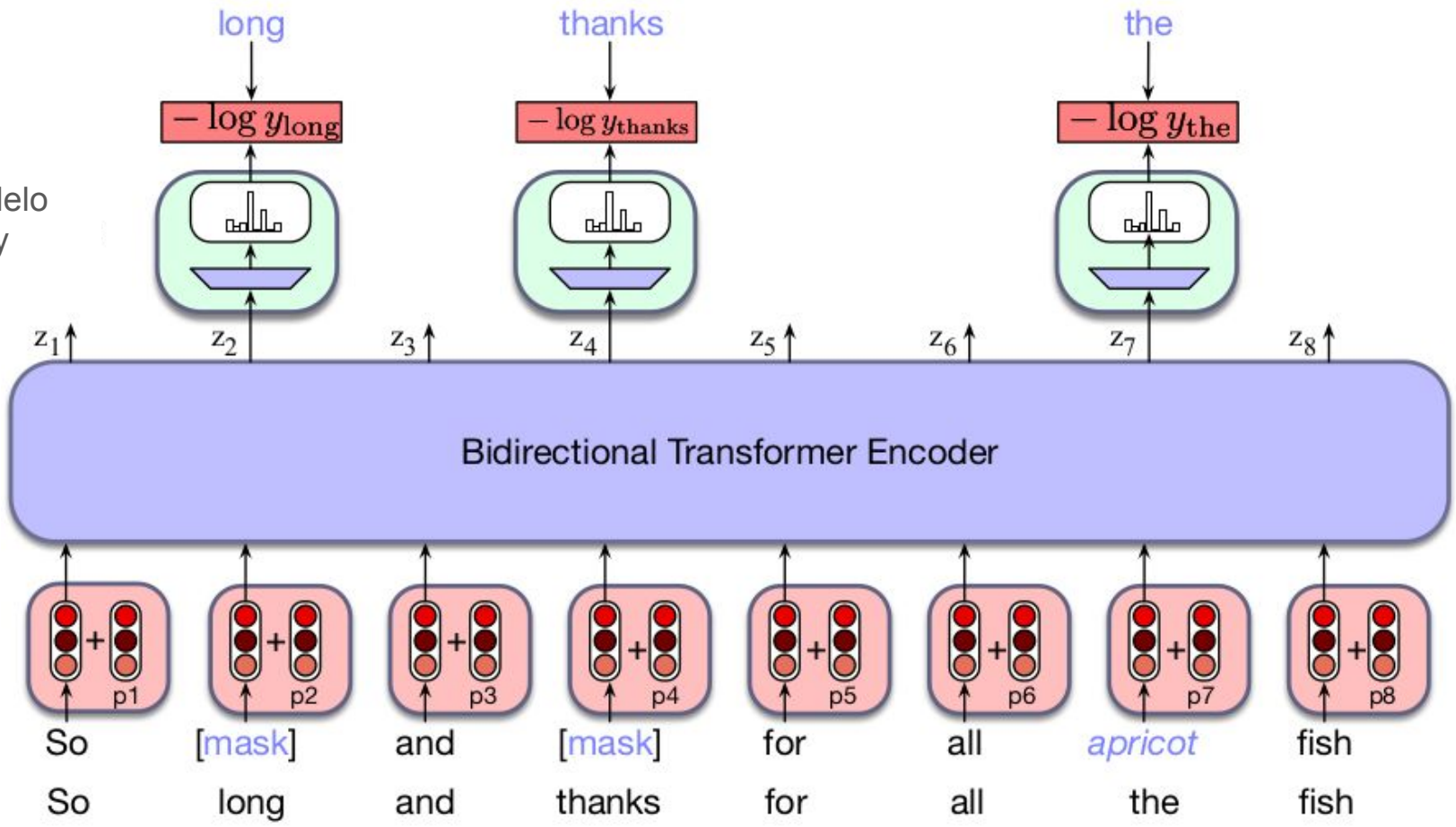
El objetivo es predecir los tokens originales (del corpus), entrenado con cross-entropy loss

Masked Language Model

loss CE

capa de modelo de lenguaje y softmax

embeddings de token y posición



Next Sentence Prediction

El objetivo de MLM es construir buenas representaciones a nivel de palabra (token)

Pero hay tareas donde lo importante es considerar la semántica de una o más oraciones globalmente

- Detección de paráfrasis
- Implicancia o *entailment*
- Contradicción

BERT agrega un segundo objetivo que apunta en esta dirección

Next Sentence Prediction

NSP es la tarea de predecir si, dados un par de oraciones, efectivamente son dos oraciones seguidas en un corpus o no

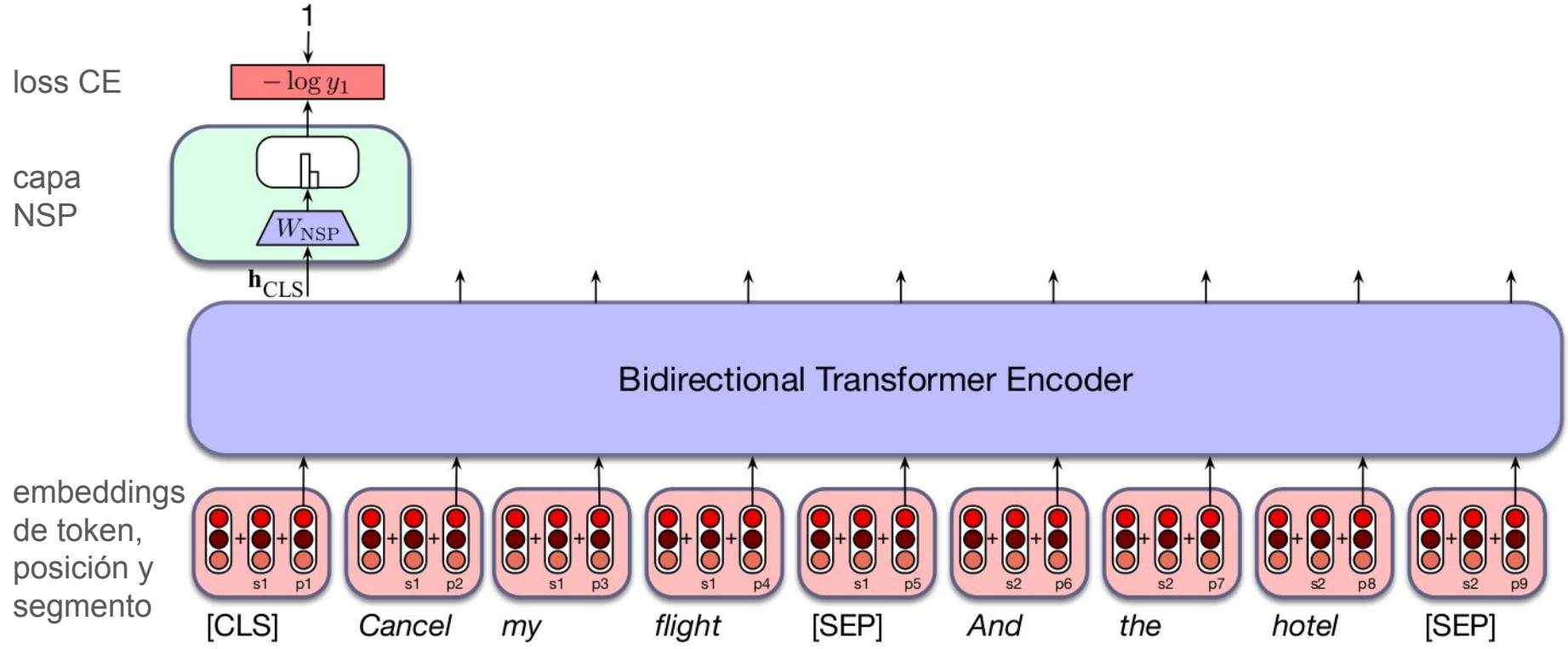
Es una tarea no supervisada que apunta a crear buenas representaciones a nivel de oración

Se sortean pares de oraciones seguidas del corpus, y luego pares aleatorios del corpus para crear ejemplos negativos

La entrada tiene la forma:

[CLS] + oración1 + [SEP] + oración2 + [SEP]

Next Sentence Prediction



Entrenamiento (preentrenamiento)

BERT original se entrenó con:

- BookCorpus (Zhu et al., 2015) (800 M), hoy no se usa por problemas de propiedad intelectual
- English Wikipedia (2.5 B)

Los LLMs actuales usan corpus órdenes de magnitud más grandes

Entrenamiento (preentrenamiento)

Se seleccionan pares de oraciones para NSP (50/50)

- El largo de ambas oraciones no supera 512 tokens
- Tokens de las oraciones son enmascarados (MLM)
- Loss combinado MLM y NSP
- 40 epochs (aprox.)

Como resultado del entrenamiento se obtienen word embeddings iniciales y parámetros del bidirectional transformer encoder, lo que permite construir representaciones contextualizadas

Luego de entrenado

Una vez entrenado, un modelo como BERT puede usarse de dos maneras

- Preentrenado (como está) para representación de embeddings contextuales
- Ajustando (fine-tuning) para tareas específicas

BERT: embeddings contextuales

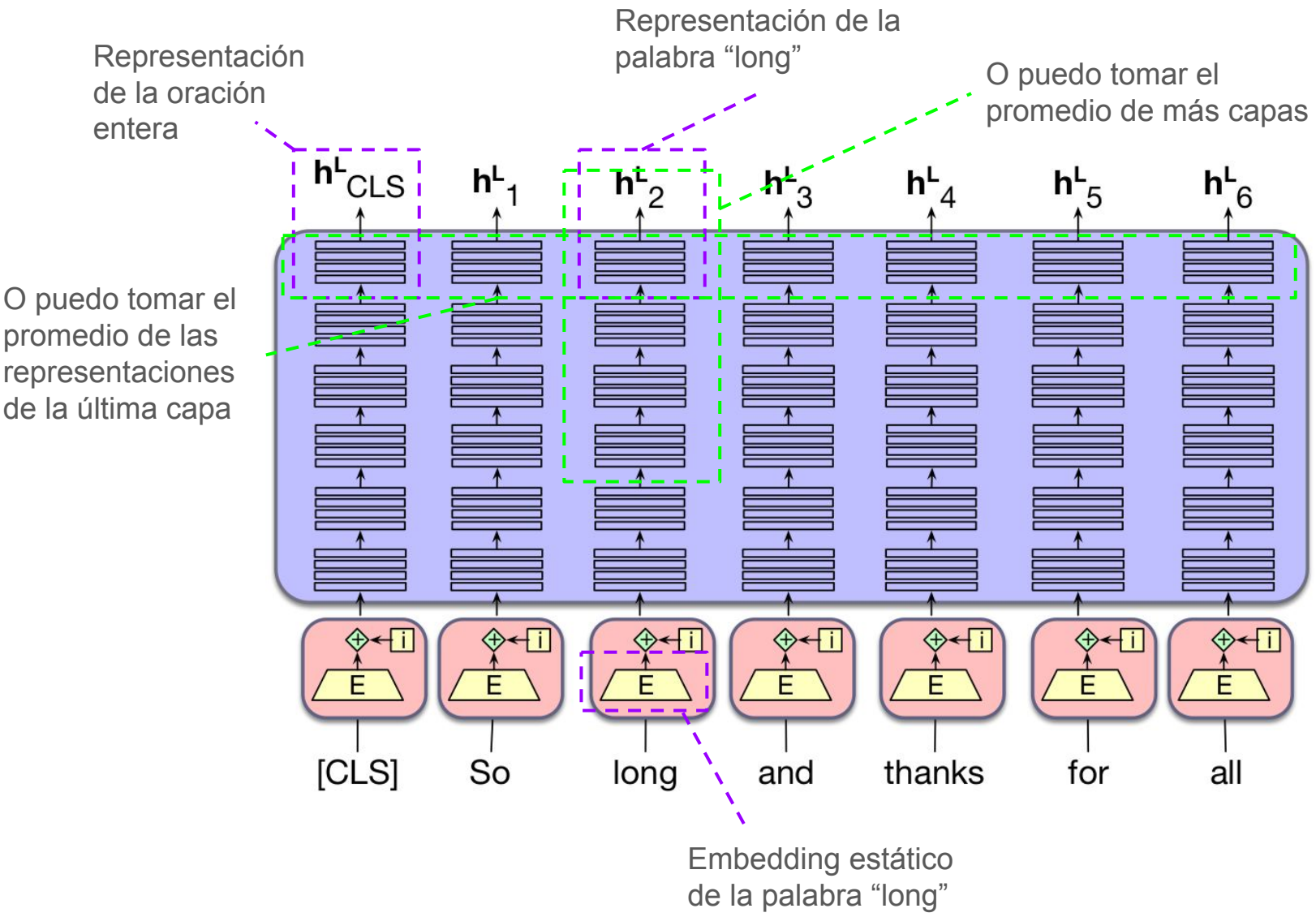
Ingresamos una secuencia (oración) de N tokens al modelo

Al ejecutarlo, obtendremos N+1 representaciones o embeddings contextuales

Internamente cada token comienza siendo un embedding estático, pero va ganando información sobre el contexto en cada capa

Finalmente, las representaciones que va tomando cada token se pueden ver como una columna de embeddings dentro del transformer

BERT: embeddings contextuales



Ejemplo: WSD

Word Sense Disambiguation (Desambiguación del Sentido de las Palabras)

Es una tarea clásica de PLN, que implica encontrar el sentido en que una palabra se usa en un contexto

Juan fue al banco$_1$ a depositar plata

Juan se sentó en el banco$_2$ de la plaza

Qué se necesita para resolver esto?

En primer lugar conocer los posibles sentidos para clasificar

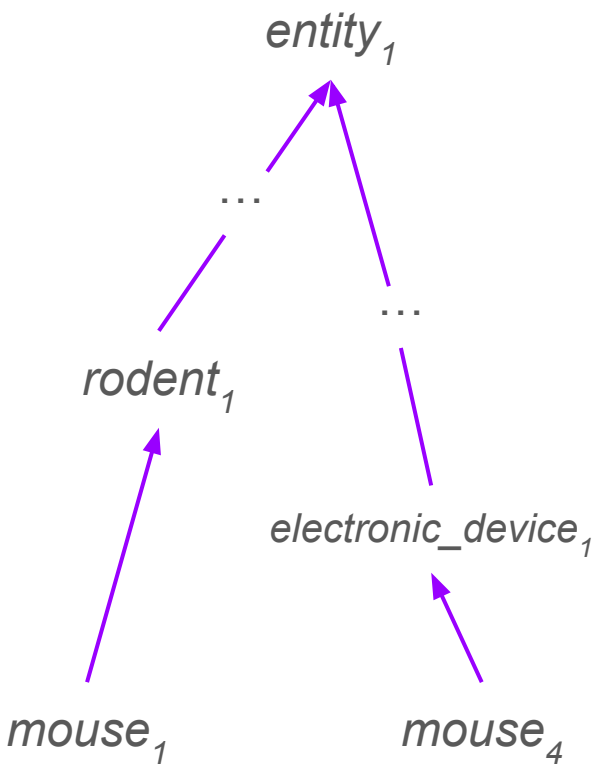
Ejemplo: WSD

WordNet (Fellbaum, 1998) es una ontología de sentidos de palabras

Inicialmente para el inglés, pero además hay versiones (incompletas) para otros idiomas, que remiten al original en inglés

- *mouse*$_1$: any of numerous small rodents typically resembling diminutive rats…
- *mouse*$_4$: a hand-operated electronic device that controls the coordinates of a cursor…
- *banco*$_1$ (es): a building in which the business of banking transacted
- *banco*$_2$ (es): a long seat for more than one person

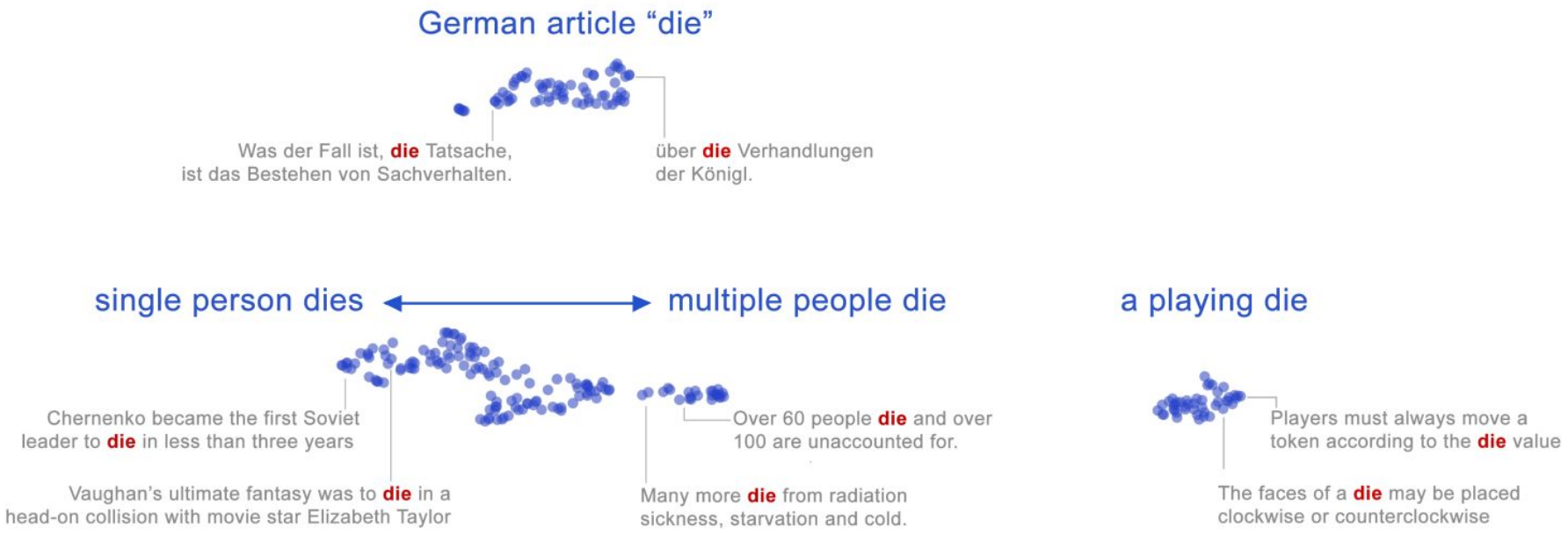
Además, los sentidos están ubicados en una ontología según la relación *"es un"* (hiperonimia)



Ejemplo: WSD

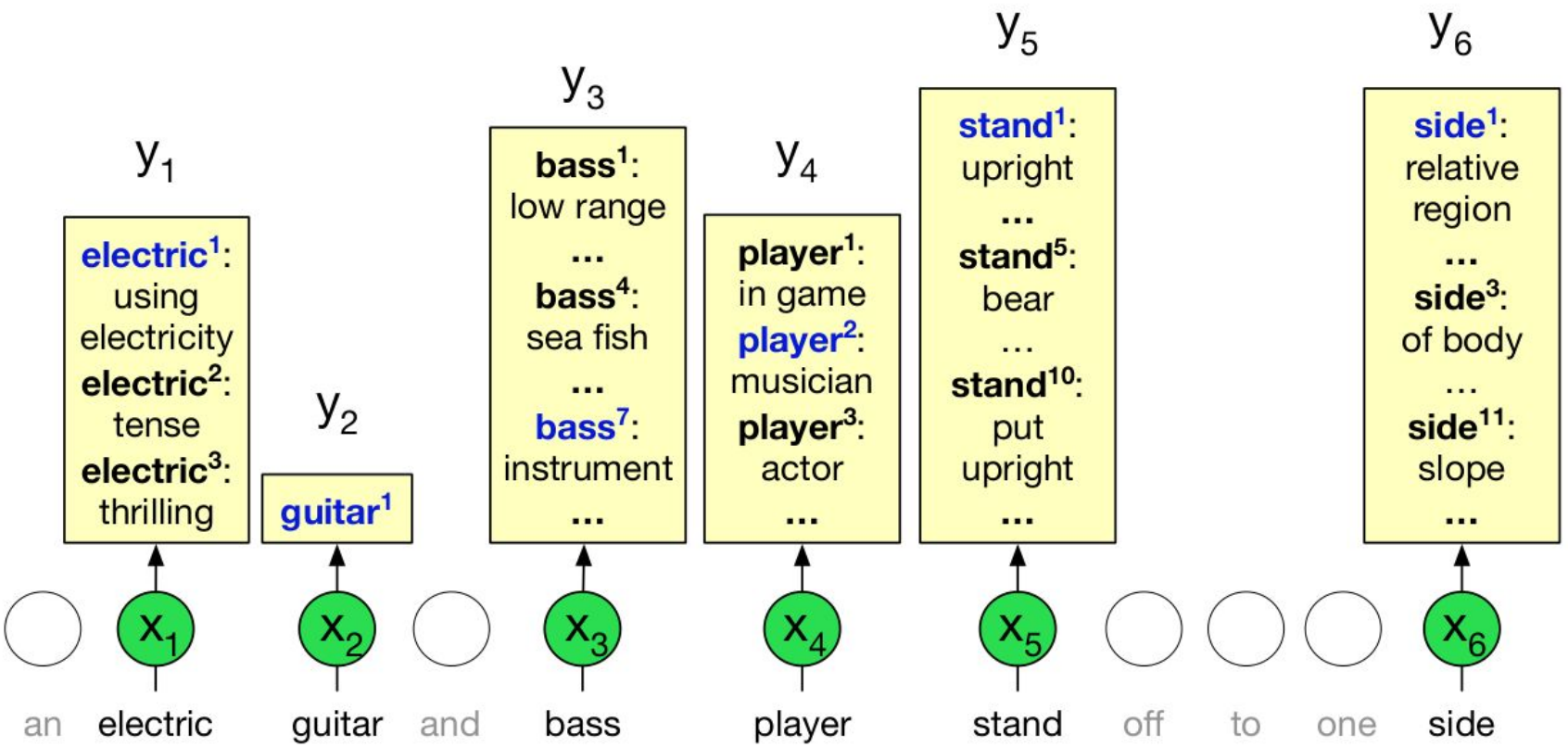
Los embeddings contextuales con BERT preentrenado ya pueden servir para hacer desambiguación semántica

Ejemplos de usos de la palabra *“die”* en inglés y alemán:



Ejemplo: WSD

Tarea “all-words WSD”: Dada una colección de sentidos discretos (como WordNet), etiquetar cada palabra de la secuencia con el sentido correcto



Ejemplo: WSD

Es una tarea muy clásica e históricamente se han buscado distintas alternativas para su solución

Pero la aparición de BERT dio por resuelto el problema usando embeddings contextuales

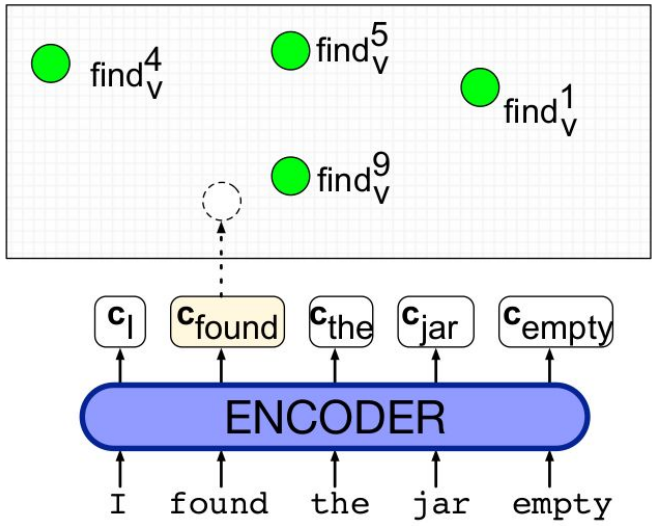
Consideremos un conjunto de training con ejemplos ya etiquetados, se los pasamos todos a BERT y obtenemos los embeddings de cada token en su contexto

Construimos un embedding v_s del sentido s

$$\mathbf{v}_s = \frac{1}{n}\sum_i \mathbf{v}_i \qquad \forall \mathbf{v}_i \in \text{tokens}(s)$$

Y luego a cada token t le asociamos el sentido más cercano según similitud coseno

$$\text{sense}(t) = \underset{s \in \text{senses}(t)}{\text{argmax}}\ \text{cosine}(\mathbf{t}, \mathbf{v}_s)$$



Entrenamiento (fine-tuning)

Al entrenamiento del BERT original le llamamos preentrenamiento, porque luego para su uso lo podemos seguir entrenado, adaptándolo a nuestra tarea

Para eso, cambiamos la última capa del modelo para adaptarla:

- Etiquetado de toda la secuencia
- Etiquetado de pares de secuencias
- Etiquetado token a token

Entrenando con un conjunto de datos particular para una nueva tarea, decimos que hacemos un ajuste o fine-tuning para la tarea

Entrenamiento (fine-tuning)

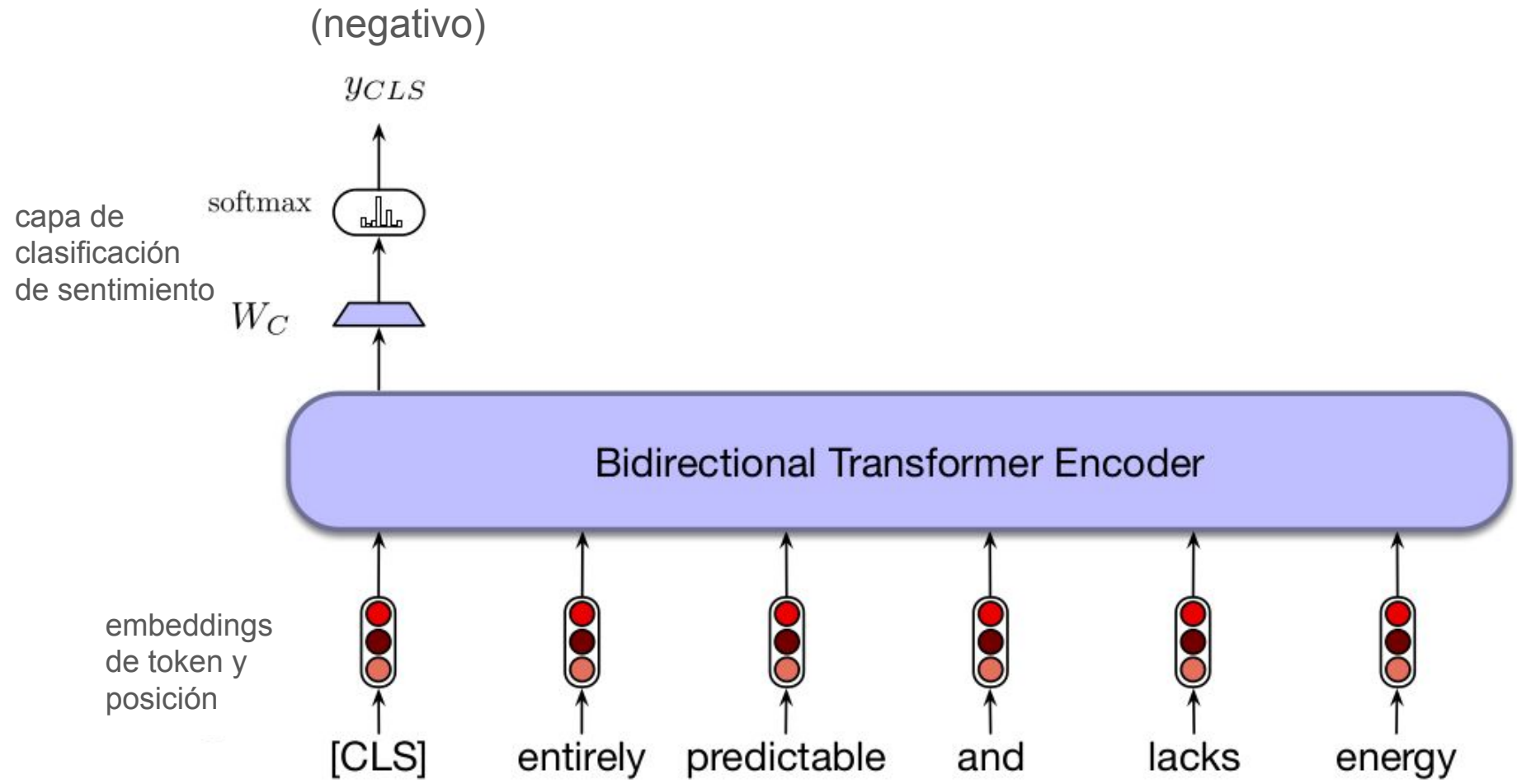
Cambiamos la última capa del modelo para adaptarla

Además de esta nueva capa, podemos (o no) seguir ajustando el resto de los pesos del transformer

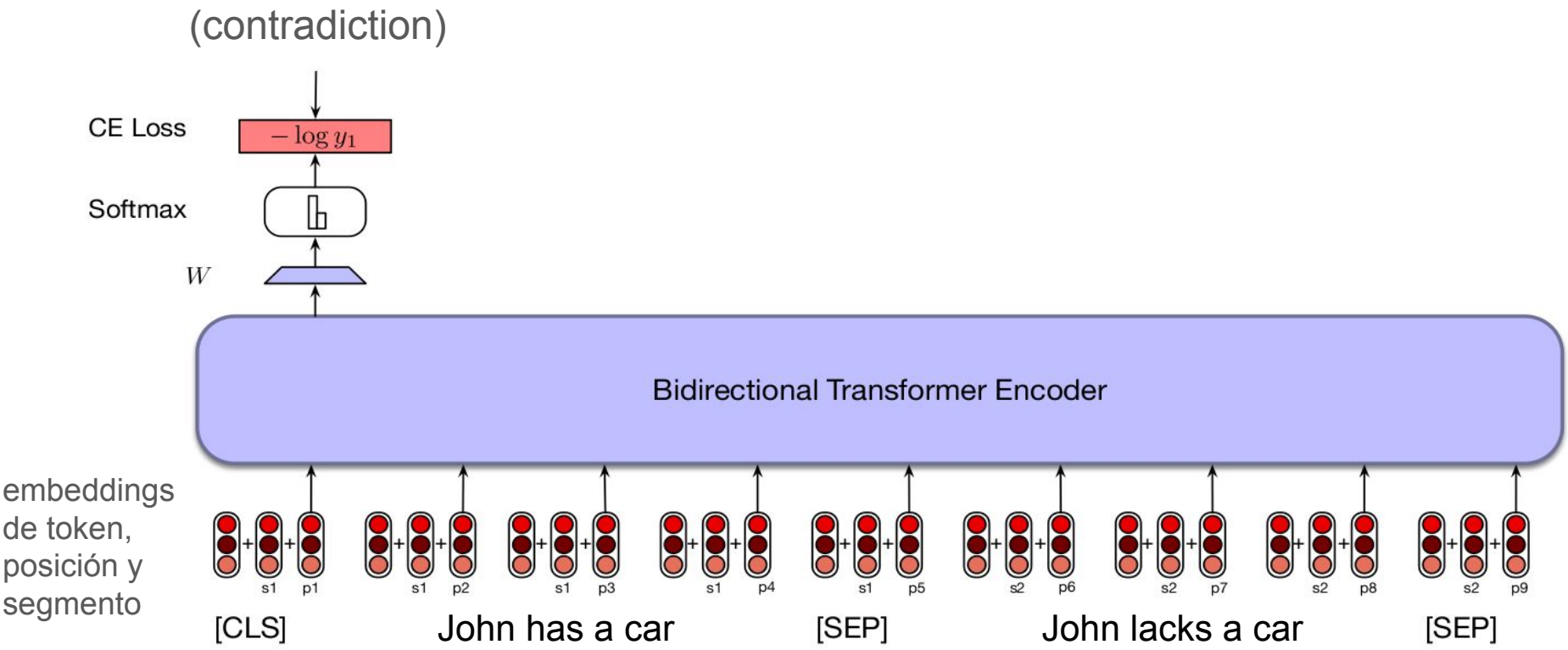
Podemos congelar los parámetros de una o más capas del transformer para que entrene más rápido

Con esto se asume que el preentrenamiento ya aprendió una buena representación del lenguaje, y solo nos falta aprender lo necesario para adaptarlo a la nueva tarea

Fine-tuning: Clasificación de secuencias



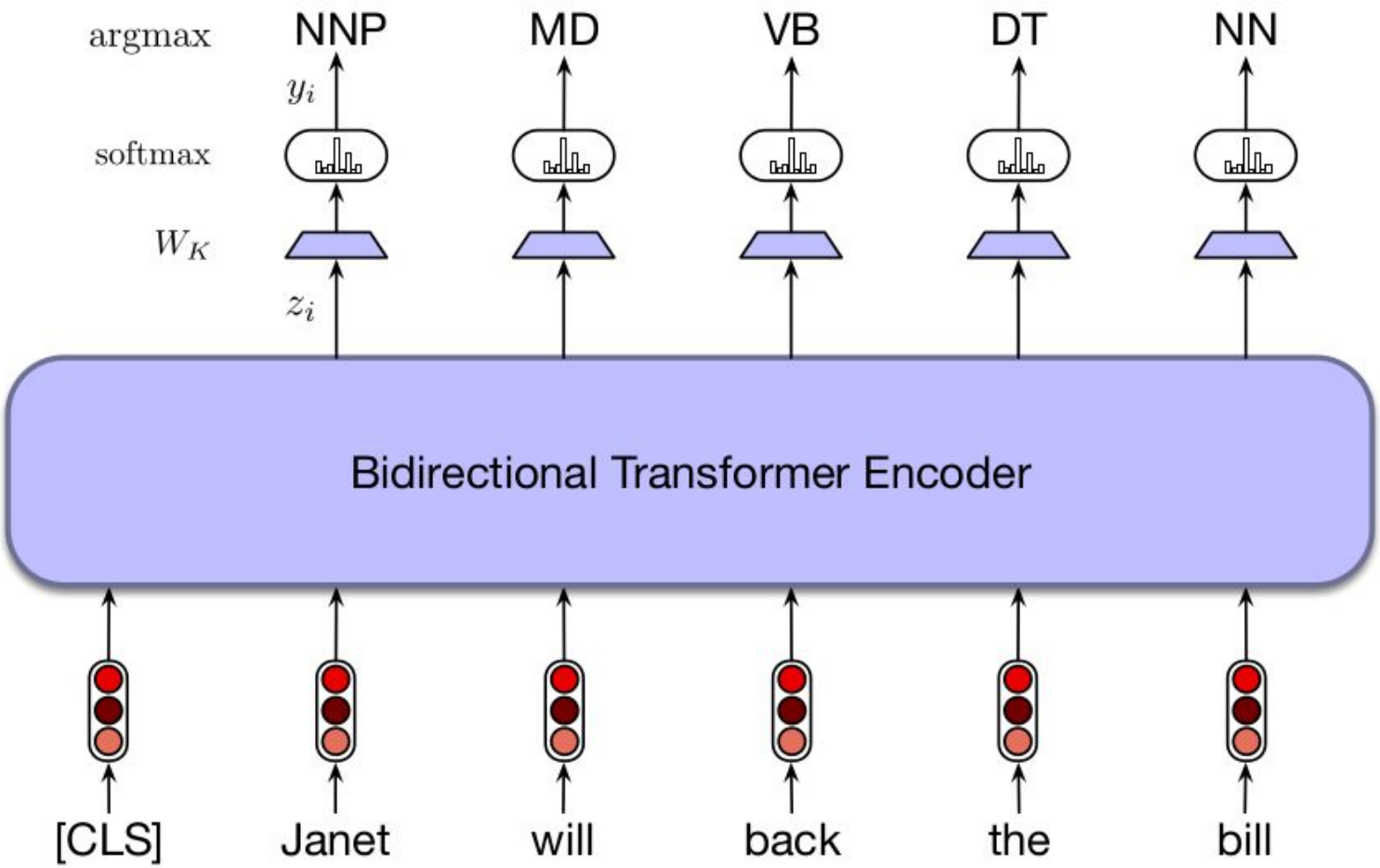
Fine-tuning: Clasificación de pares de secuencias



Fine-tuning: Etiquetado secuencial

capa de POS tagging

embeddings de token y posición



Ejemplo: NER

Named Entity Recognition (Reconocimiento de Entidades con Nombre)

Se le llama entidad con nombre a todo lo que puede ser referenciado mediante un nombre propio

En esta tarea, se deben identificar segmentos contiguos (*spans*) de texto que representan entidades con nombre

Y clasificarlos según su tipo: persona, lugar, organización, otras…

Ejemplo: NER

¿Por qué es difícil?

Spans de texto iguales pueden referir a entidades de diferente tipo

[$_{PER}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{ORG}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{LOC}$ Washington] for what may well be his last state visit.
In June, [$_{GPE}$ Washington] passed a primary seatbelt law.

Y si no tuviéramos ayuda de las mayúsculas, también habría que distinguir los usos literales

LA [$_{ORG}$ CASA BLANCA] EMITIÓ UNA DECLARACIÓN

YO VIVO EN UNA CASA BLANCA

Ejemplo: NER

Para representar el texto en esta tarea se utilizan notaciones especiales, la más común es BIO (Begin-Inside-Outside)

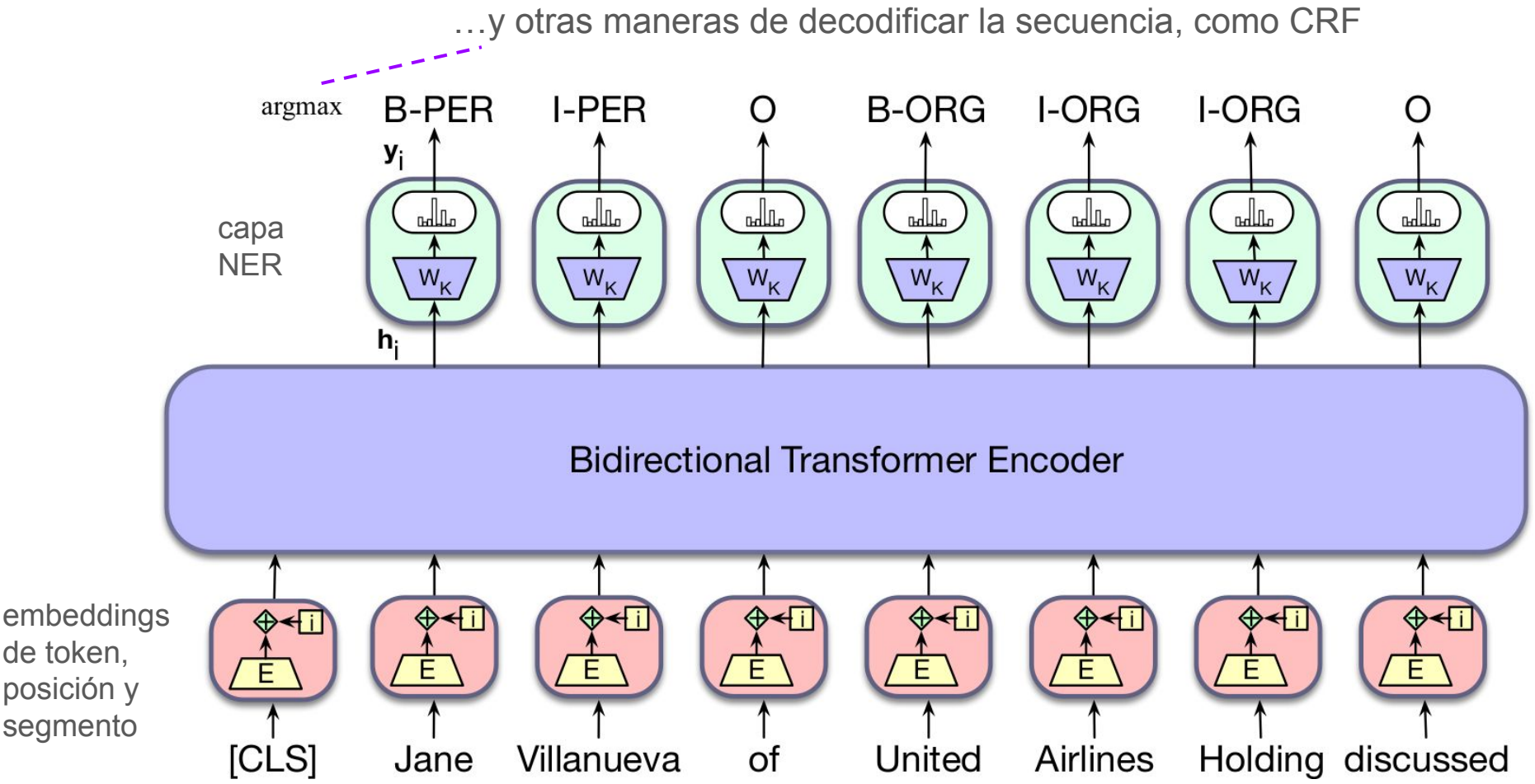
[$_{ORG}$Ministerio de Salud] omombe'u ko'ã káso malaria ojuhúva importado [$_{LOC}$Guinea Ecuatorial] guive ha oîma jesareko ohapejokóvo jeipyso .

[$_{PER}$Ministra de Hacienda Lea Giménez] he'i oñepromulga léi capitalidad ary 2014

Ministerio	B-ORG
de	I-ORG
Salud	I-ORG
omombe'u	O
ko'ã	O
káso	O
malaria	O
ojuhúva	O
importado	O
Guinea	B-LOC
Ecuatorial	I-LOC
guive	O
ha	O
…	

Competencia GUA-SPA@IberLEF2023 de detección de code-switching y NER en guaraní y español
https://www.fing.edu.uy/inco/grupos/pln/guaspa/

Ejemplo: NER


…y otras maneras de decodificar la secuencia, como CRF
argmax
B-PER
I-PER
O
B-ORG
I-ORG
I-ORG
O
y_i
capa NER
W_K
h_i
Bidirectional Transformer Encoder
embeddings de token, posición y segmento
E
i
[CLS]
Jane
Villanueva
of
United
Airlines
Holding
discussed

Ejemplo: NER

Los modelos tipo BERT utilizan algún esquema de tokenización subpalabra

Eso puede romper las entidades con nombre

Mt. *Sanitas is in Sunshine Canyon .*
B-LOC I-LOC O O B-LOC I-LOC O

'Mt', '.', 'San', '##itas', 'is', 'in', 'Sunshine', 'Canyon' '.'

Hay que definir una forma de etiquetar los tokens subpalabra

Y al obtener el resultado, volver a rearmar las entidades que puedan haber quedado partidas

Por ejemplo: asignar la misma etiqueta a todas las subpalabras en entrenamiento, pero al obtener la salida considerar solo el tag del primer token

Ejemplo: NER

La evaluación de NER se suele hacer con precisión, recall y F1

Pero como son spans de texto, puede que algunas palabras queden afuera

- Estricto vs. Relajado

Además, tenemos la tarea de encontrar el span correctamente, y además la de predecir la categoría correcta

- Etiquetado vs. No etiquetado

Sabores de BERT

BERT es el modelo original propuesto por Devlin et al. (2019), entrenado para el inglés

Pero con el tiempo han aparecido muchos modelos más, variantes de BERT, que lo mejoran, modifican el entrenamiento, agregan idiomas, etc.

- RoBERTa: En inglés, más datos, sin NSP (Liu et al., 2019)
- XLM-RoBERTa: Versión multilingüe de RoBERTa (Conneau et al,, 2019)
- AlBERT: Mismos datos que BERT, matrices factorizadas, es más rápido
- BETO: BERT entrenado con datos en español (Cañete et al., 2020)
- RoBERTuito: Entrenado con tweets en español, para análisis de sentimiento
- ROUBERTa: Entrenado con texto de prensa uruguayo! (Filevich et al., 2024)
- SentenceBERT: Modificación para hacer buenos embeddings de oraciones

…

Sabores de BERT

Comparación de BERT y XLM-RoBERTa

	BERT	XLM-RoBERTa
Idioma	Solo inglés	Multilingüe (100 idiomas)
Vocabulario	30.000 tokens, WordPiece	250.000 tokens, SentencePiece
Corpus	3.3 B tokens	2.5 TB, CommonCrawl
Tamaño de embedding	768	1024
Capas	12 bloques transformer con 12 cabezales de atención cada uno	24 bloques transformer con 16 cabezales de atención cada uno
Entrenamiento	MLM, NSP	MLM
Contexto máximo	512	512
Total parámetros	100 M	550 M

GPT3 (2020) ya tenía 175B parámetros!

Evaluación

Dada la diversidad de modelos de lenguaje tipo BERT, cómo sabemos cuál es mejor?

Existen benchmarks de evaluación que contienen varias tareas y se usan para comparar diferentes modelos

General Language Understanding Evaluation (GLUE) fue el primero de estos benchmarks, incluyendo tareas a nivel de oración o pares de oraciones

- Detección de paráfrasis
- Análisis de sentimiento
- Preguntas y respuestas
- Inferencia
- Otras…

GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding
Wang, Singh, Michael, Hill, Levy, Bowman (2018)

Evaluación

Luego aparecen otros benchmarks

- GLUES para español
- XGLUE multilingüe
- SuperGLUE incorpora tareas de razonamiento con más texto

Siempre son tareas discriminativas (no generativas), por eso los modelos generativos como GPT no aparecen en estos rankings (les suele ir peor)

Hay otros benchmarks para tareas generativas

Evaluación - GLUE Leaderboard

Rank	Name	Model	Score	CoLA	SST-2	MRPC	STS-B	QQP	MNLI-m	MNLI-mm	QNLI	RTE	WNLI	AX
1	Microsoft Alexander v-team	Turing ULR v6	91.3	73.3	97.5	94.2/92.3	93.5/93.1	76.4/90.9	92.5	92.1	96.7	93.6	97.9	55.4
2	JDExplore d-team	Vega v1	91.3	73.8	97.9	94.5/92.6	93.5/93.1	76.7/91.1	92.1	91.9	96.7	92.4	97.9	51.4
3	Microsoft Alexander v-team	Turing NLR v5	91.2	72.6	97.6	93.8/91.7	93.7/93.3	76.4/91.1	92.6	92.4	97.9	94.1	95.9	57.0
4	DIRL Team	DeBERTa + CLEVER	91.1	74.7	97.6	93.3/91.1	93.4/93.1	76.5/91.0	92.1	91.8	96.7	93.2	96.6	53.3
5	ERNIE Team - Baidu	ERNIE	91.1	75.5	97.8	93.9/91.8	93.0/92.6	75.2/90.9	92.3	91.7	97.3	92.6	95.9	51.7
6	AliceMind & DIRL	StructBERT + CLEVER	91.0	75.3	97.7	93.9/91.9	93.5/93.1	75.6/90.8	91.7	91.5	97.4	92.5	95.2	49.1
...														
20	Shiwen Ni	ELECTRA-large-M (bert4keras)	88.3	69.3	95.8	92.2/89.6	91.2/91.1	75.1/90.5	91.1	90.9	93.8	87.9	91.8	48.2
21	Facebook AI	RoBERTa	88.1	67.8	96.7	92.3/89.8	92.2/91.9	74.3/90.2	90.8	90.2	95.4	88.2	89.0	48.7
22	Microsoft D365 AI & MSR AI	MT-DNN-ensemble	87.6	68.4	96.5	92.7/90.3	91.1/90.7	73.7/89.9	87.9	87.4	96.0	86.3	89.0	42.8
23	GLUE Human Baselines	GLUE Human Baselines	87.1	66.4	97.8	86.3/80.8	92.7/92.6	59.5/80.4	92.0	92.8	91.2	93.6	95.9	-
...														
48	USCD-AI4Health Team	CERT	80.7	58.9	94.6	89.8/85.9	87.9/86.8	72.5/90.3	87.2	86.4	93.0	71.2	65.1	39.6
49	Jacob Devlin	BERT: 24-layers, 16-heads, 1024-hidden	80.5	60.5	94.9	89.3/85.4	87.6/86.5	72.1/89.3	86.7	85.9	92.7	70.1	65.1	39.6
50	Chen Qian	KerasNLP XLM-R	80.4	56.3	96.1	89.8/86.3	88.4/87.7	72.3/89.0	87.7	87.1	92.8	69.2	65.1	40.6
...														

https://gluebenchmark.com/leaderboard

Tipos de problemas

1 palabra (o un par) → 1 categoría

frío y *caliente* son sinónimos?
antónimos? **MLP**

n palabras → 1 categoría (k posibles)

este tweet tiene sentimiento positivo?
este tweet es un chiste?
MLP, CNN, RNN, BERT

n palabras → 0..k categorías

de qué temas habla este texto?
qué emociones presenta este tweet?
MLP, CNN, RNN, BERT

n palabras → n categorías

POS-tagging, NER,
chunking, parsing
CNN, RNN, BERT

n palabras → m palabras

traducción automática
respuestas a preguntas
resúmenes automáticos
...
Encoder-Decoder (RNN, Transformer)

Bibliografía

- Jurafsky and Martin 3rd edition. Capítulo 11. https://web.stanford.edu/~jurafsky/slp3/

- Papers...