Transformers: BERT

Orlando Ramos Flores

Contenido

- BERT
  - ¿Qué es BERT?
  - Motivación
  - Masked Language Model
  - BERT
  - Tokenización

BERT

¿Qué es BERT?

- BERT (Bidirectional Encoder Representations from Transformers) es nuevo modelo de representación del lenguaje, y está diseñado para pre-entrenar representaciones bidireccionales profundas a partir de texto sin etiquetar mediante el condicionamiento conjunto del contexto izquierdo y derecho en todas las capas.
- Como resultado, el modelo BERT pre-entrenado se puede ajustar con solo una capa de salida adicional para crear modelos de última generación para una amplia gama de tareas.

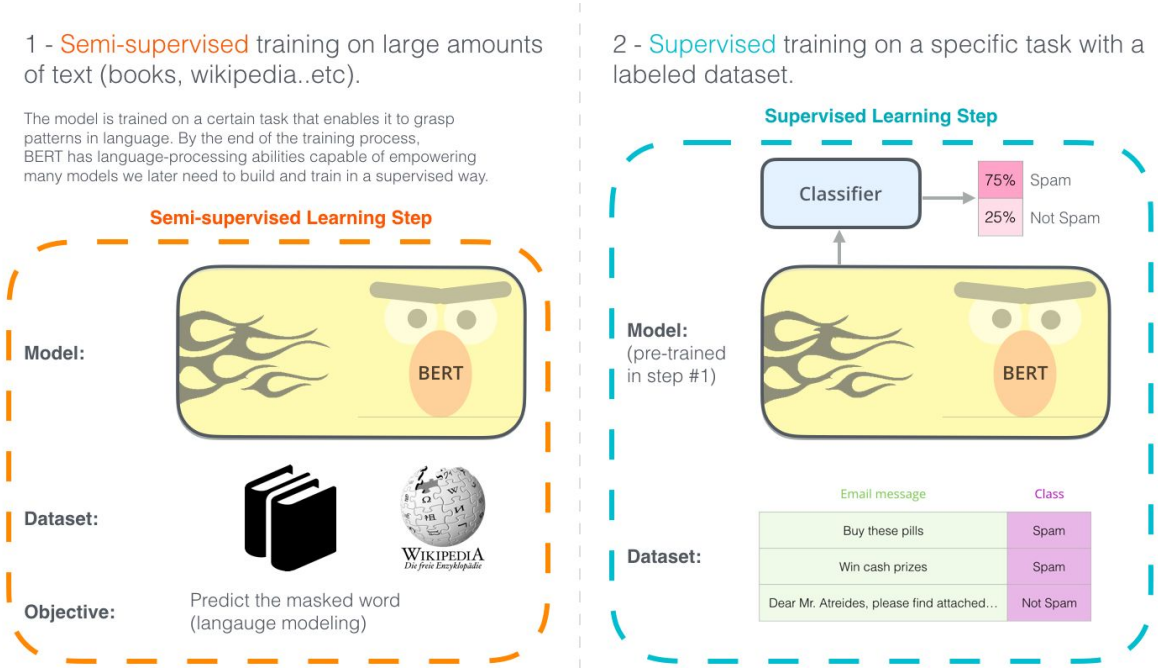
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

BERT: Base y Large

- Sea L el número de capas (bloques de Transformers). El tamaño del estado oculto denotado por H, y el número de cabezas de self-attention como A.
- $BERT_{BASE}$
  - L = 12
  - H = 768
  - A = 12
  - Parámetros = 110M
- $BERT_{LARGE}$
  - L = 24
  - H = 1024
  - A = 16
  - Parámetros = 340M

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

BERT

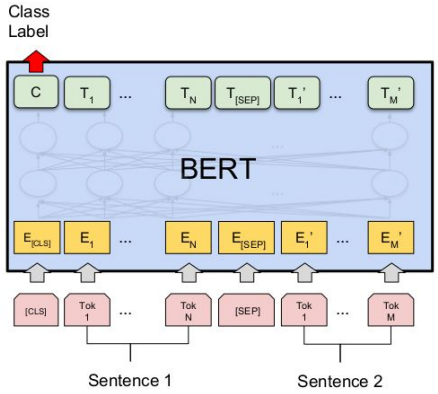


Los dos pasos de cómo se desarrolla BERT. Se puede descargar el modelo pre-entrenado en el paso 1 (entrenado con datos no anotados) y solo preocuparse por ajustarlo para el paso 2.

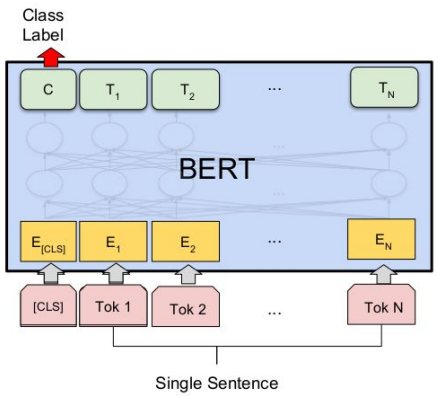
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fine-tuning BERT

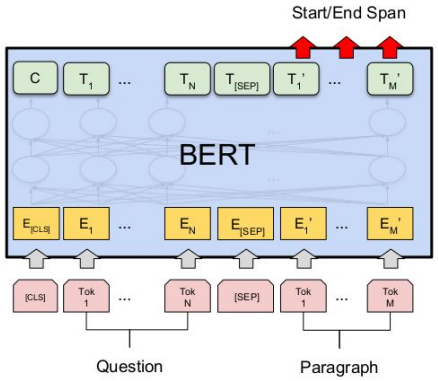
- Obtiene nuevos resultados de última generación en once tareas de procesamiento de lenguaje natural,
- Incluido el aumento de la puntuación GLUE al 80,5 % (7,7 % de mejora absoluta),
- La precisión de MultiNLI al 86,7 % (4,6 % de mejora absoluta),
- SQuAD v1.1 Question & Answering Test F1 a 93.2 (1.5 puntos de mejora absoluta) y
- SQuAD v2.0 Test F1 a 83.1 (5.1 puntos de mejora absoluta).



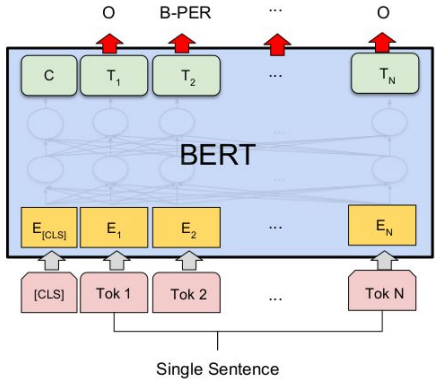
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



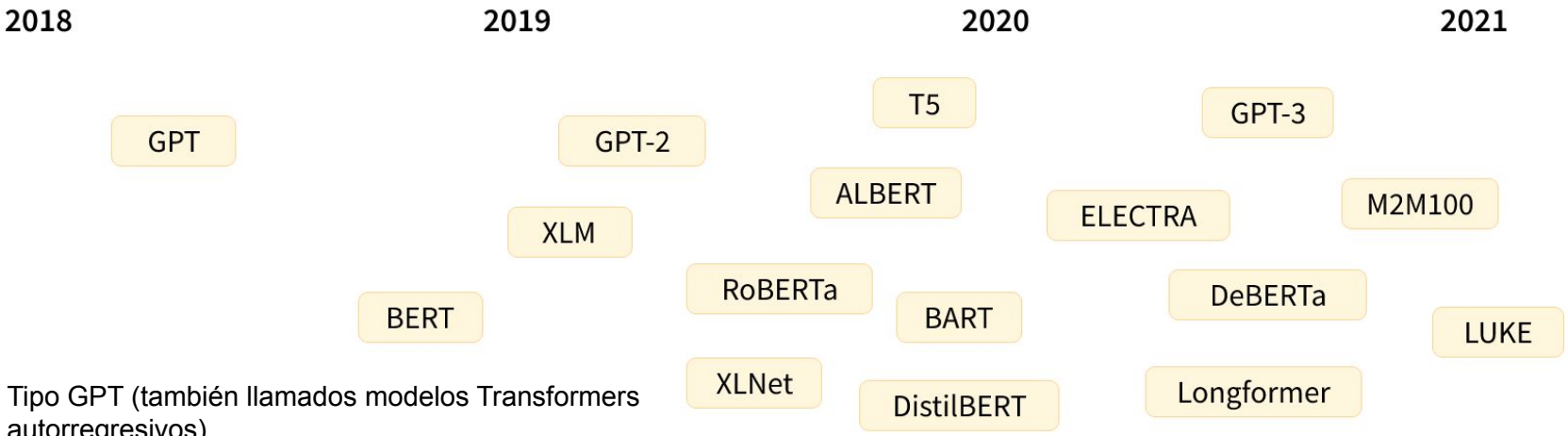
(c) Question Answering Tasks: SQuAD v1.1



(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Modelos de Transformers: Historia



- Tipo GPT (también llamados modelos Transformers autorregresivos).
- Tipo BERT (también llamados modelos Transformers de codificación automática).
- Tipo BART/T5 (también llamados modelos de Transformers de secuencia a secuencia)

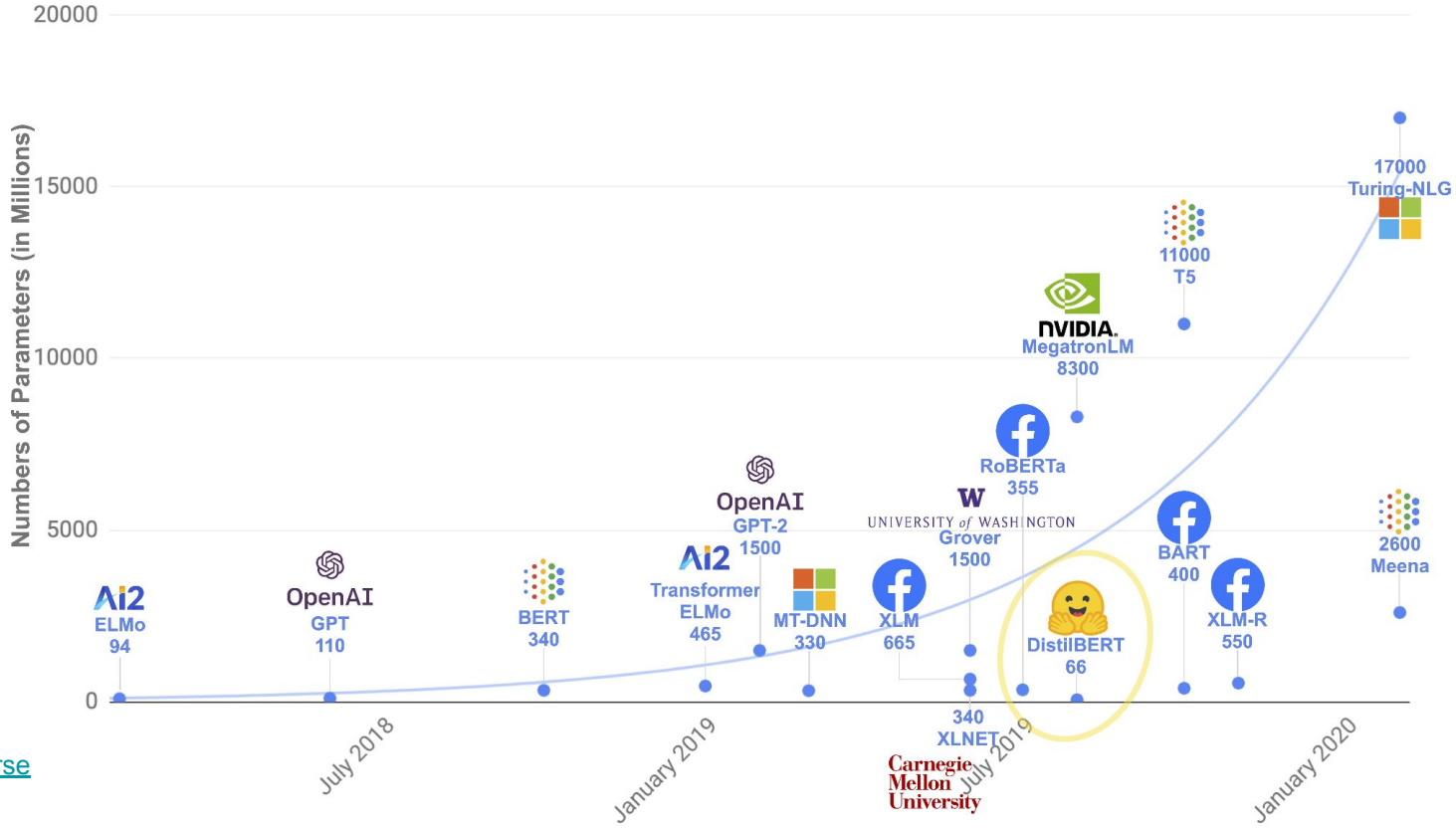
Hugging Face Course

Modelos de Transformers: Historia

- Todos los modelos de Transformer mencionados anteriormente (GPT, BERT, BART, T5, etc.) han sido entrenados como **modelos de lenguaje**.
- Esto significa que han sido entrenados sobre grandes cantidades de texto sin procesar de manera auto supervisada.
- El aprendizaje autosupervisado es un tipo de entrenamiento en el que el objetivo se calcula automáticamente a partir de las entradas del modelo.
- Significa que no se necesitan humanos para etiquetar los datos

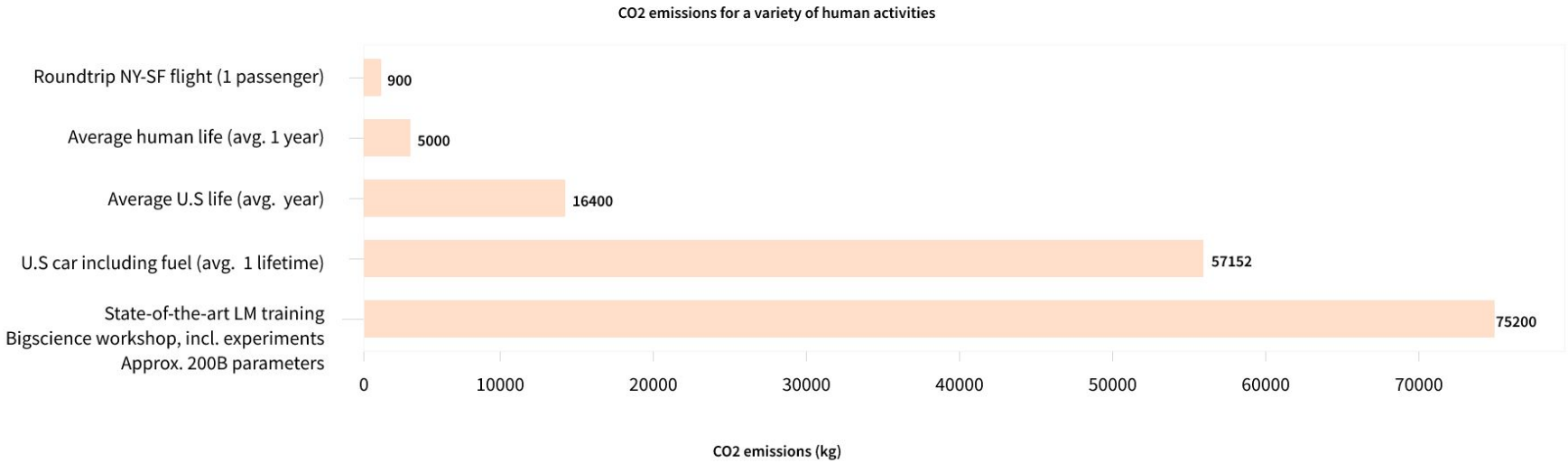
Hugging Face Course

Transformers



[Hugging Face Course]

Transformers: Huella de carbono



Hugging Face Course
The carbon footprint of Transformers

Motivación

BERT: Motivación

- Hay dos estrategias existentes para aplicar **representaciones de lenguaje previamente entrenadas** a *downstream tasks*[1]: featured-based (basadas en características) y fine-tuning (de ajuste fino).
- El enfoque featured-based con ELMo utiliza arquitecturas específicas de tareas que incluyen las representaciones pre-entrenadas como características adicionales.
- El enfoque fine-tuning tal como Generative Pre-trained Transformer (OpenAI GPT) presenta parámetros mínimos específicos de la tarea y se entrena en *downstream tasks*[1] simplemente ajustando todos los parámetros pre-entrenados.

1. Downstream tasks: en el campo del PLN se llaman a aquellas tareas de aprendizaje supervisado que utilizan un modelo o componente pre-entrenado

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

BERT: Motivación

- Los dos enfoques comparten la misma función objetivo durante el pre-entrenamiento, donde usan **modelos de lenguaje unidireccionales** para aprender representaciones generales del lenguaje.
- En el paper BERT argumentan que tales técnicas restringen el poder de las representaciones pre-entrenadas, especialmente para los enfoques de fine-tuning.
- La principal limitación es que los modelos de lenguaje estándar son unidireccionales, y esto limita la elección de arquitecturas que se pueden usar durante el pre-entrenamiento.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

BERT: Motivación

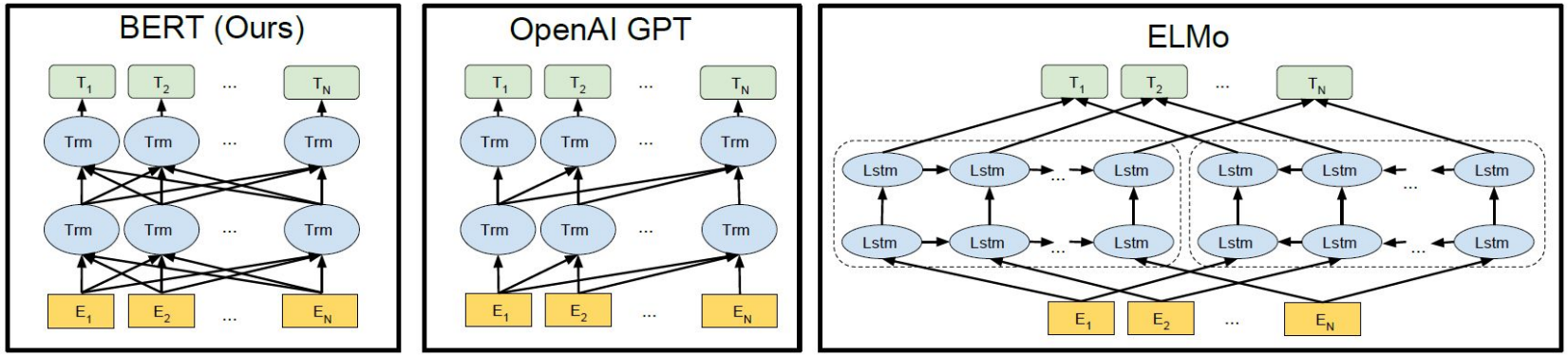


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Masked Language Model

Masked Language Model (MLM)

- Intuitivamente, es razonable creer que un modelo bidireccional profundo es estrictamente más poderoso que un modelo de izquierda-a-derecha o que uno con concatenación superficial de un modelo de izquierda-a-derecha y de derecha-a-izquierda.
- Desafortunadamente, los modelos de lenguaje condicional estándar solo se pueden entrenar de izquierda-a-derecha o de derecha-a-izquierda, ya que el condicionamiento bidireccional permitiría que cada palabra se “viera a sí misma” indirectamente, y el modelo podría predecir trivialmente la palabra objetivo en un formato de contexto de varias capas.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Masked Language Model (MLM)

- Entonces, para entrenar una representación bidireccional profunda, simplemente se enmascara algún porcentaje de los tokens de entrada al azar y luego se predicen esos tokens enmascarados.
- Este procedimiento se refiere como un model “masked LM” (MLM), aunque a menudo se lo denomina tarea *Cloze* en la literatura.
- En este caso, los vectores ocultos finales correspondientes a los tokens de máscara se introducen en un softmax de salida sobre el vocabulario, como en un LM estándar.
- En todos los experimentos, enmascaran el 15 % de todas las piezas de palabras en cada secuencia al azar.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
Taylor, W. L. (1953). “Cloze procedure”: A new tool for measuring readability. Journalism quarterly, 30(4), 415-433.

Masked Language Model (MLM)

Alaska		York
Alaska is		New York
Alaska is about		than New York
Alaska is about twelve		larger than New York
Alaska is about twelve times		times larger than New York
Alaska is about twelve times larger		twelve times larger than New York
Alaska is about twelve times larger than		about twelve times larger than New York
Alaska is about twelve times larger than New		is about twelve times larger than New York
Alaska is about twelve times larger than New York		Alaska is about twelve times larger than New York
Left-to-right prediction	**Word prediction using context from only one side**	Right-to-left prediction

Masked Language Model (MLM)

Word prediction using context from both sides (e.g. BERT)

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

Alaska is about twelve times larger than New York

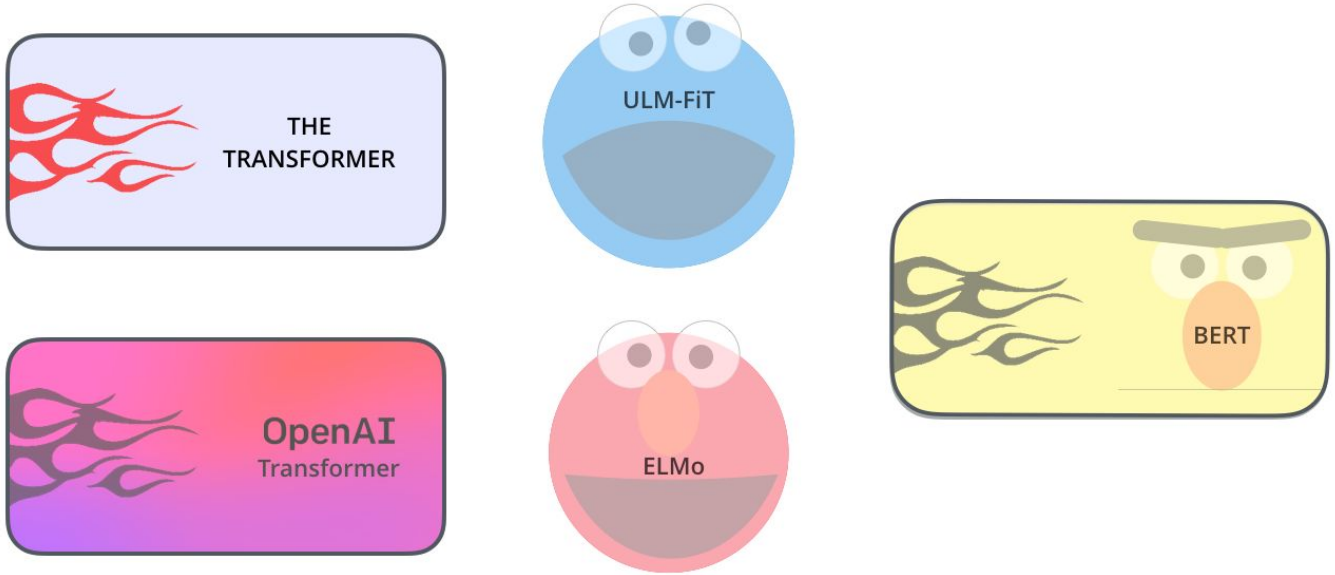
Alaska is about twelve times larger than New York

MLM: Procedimiento

- Suponiendo que la oración sin etiquetar es: *mi perro es peludo*, y durante el procedimiento de enmascaramiento aleatorio se elige el 4º token (que corresponde a *peludo*), en el procedimiento de enmascaramiento se tiene:
  - El **80%** de las veces: Reemplazaron la palabra con el token [MASK], por ejemplo: *mi perro es peludo* → mi perro es [MASK]
  - **10%** de las veces: Reemplazaron la palabra con una palabra aleatoria, por ejemplo: *mi perro es peludo* → mi perro es manzana
  - **10%** de las veces: Mantuvieron la palabra sin cambios, por ejemplo: *mi perro es peludo* → *mi perro es peludo*. El propósito de esto es sesgar (bias) la representación hacia la palabra real observada.
- La ventaja de este procedimiento es que el encoder del Transformer no sabe qué palabras se le pedirá que prediga o cuáles han sido reemplazadas por palabras aleatorias, por lo que se ve obligado a mantener una representación contextual distribucional de cada token de entrada.

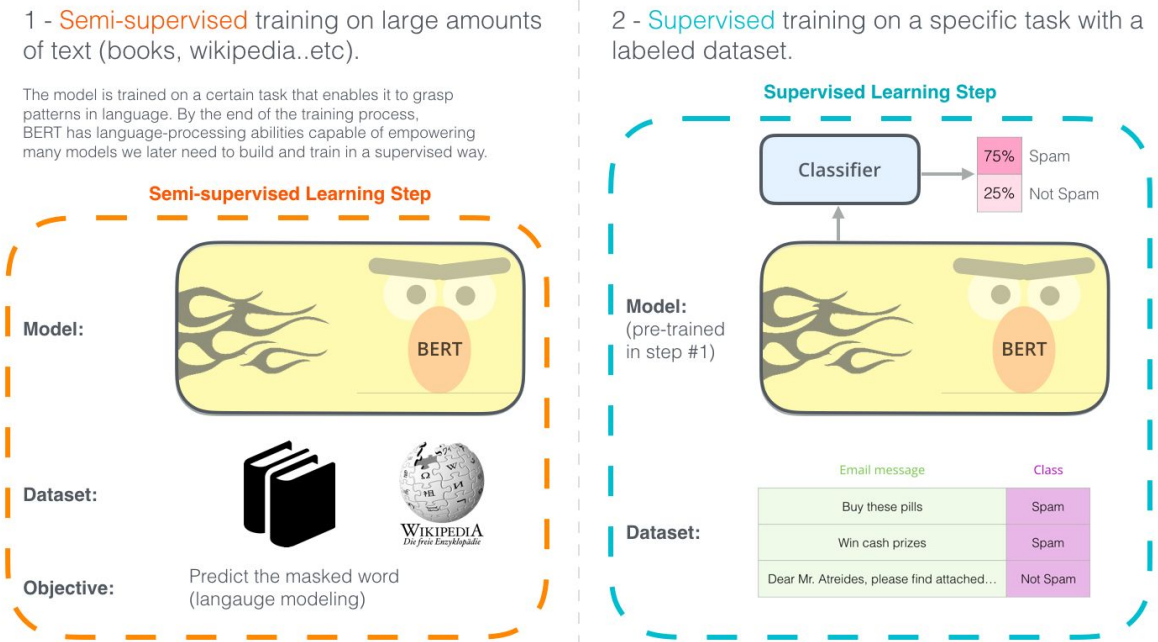
BERT

BERT



Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

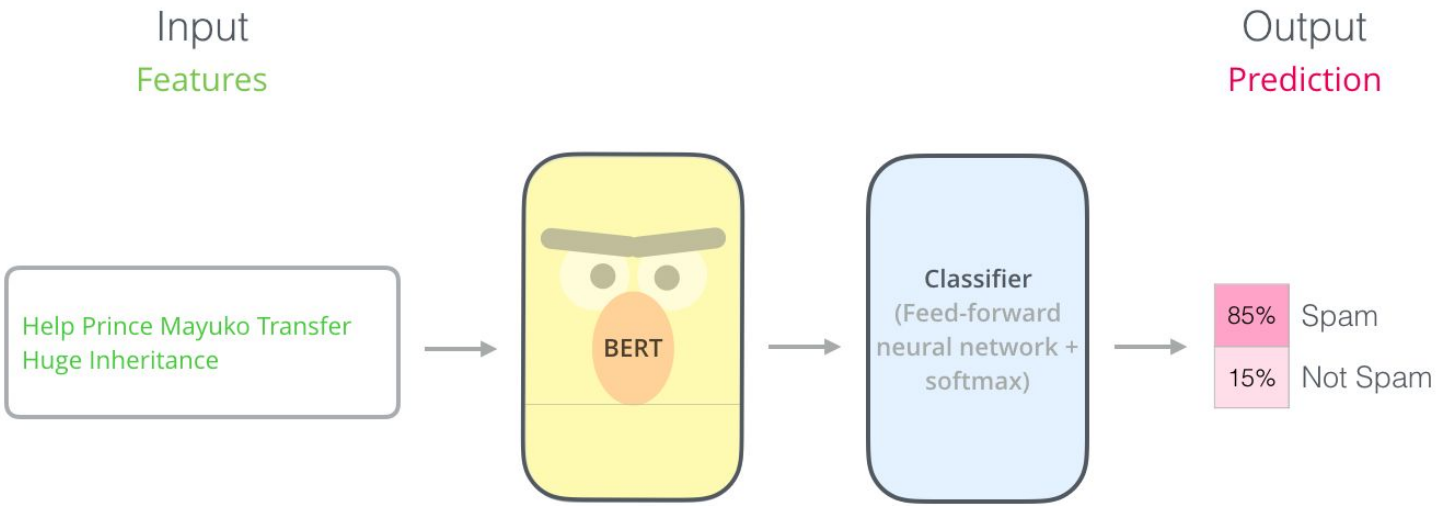
BERT



Los dos pasos de cómo se desarrolla BERT. Se puede descargar el modelo pre-entrenado en el paso 1 (entrenado con datos no anotados) y solo preocuparse por ajustarlo para el paso 2.

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

BERT. Ejemplo: Clasificación de oraciones



La forma más directa de usar BERT es usarlo para clasificar una sola pieza de texto. Para entrenar un modelo de este tipo, principalmente se tiene que entrenar el clasificador, con cambios mínimos en el modelo BERT durante la fase de entrenamiento. Este proceso de entrenamiento se llama Fine-Tuning (Ajuste Fino) y tiene sus raíces en el Aprendizaje Secuencial Semi-supervisado y ULMFiT.

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

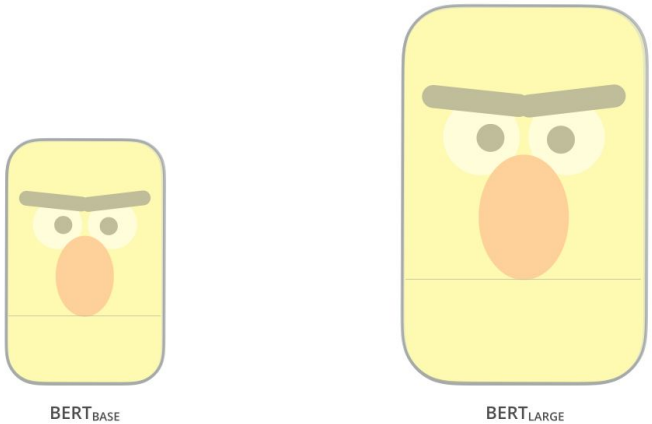
Conjunto de datos

Email message	Class
Buy these pills	Spam
Win cash prizes	Spam
Dear Mr. Atreides, please find attached…	Not Spam

Para este ejemplo de clasificador de spam, el conjunto de datos etiquetado sería una lista de mensajes de correo electrónico y una etiqueta ("spam" o "no spam" para cada mensaje).

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

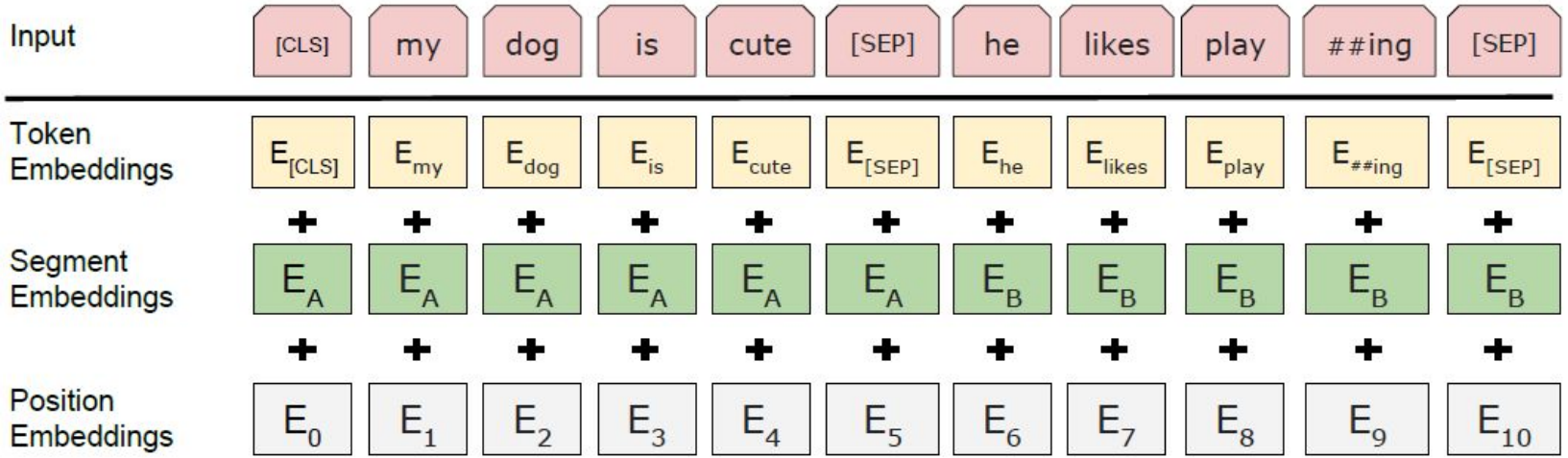
Arquitectura del modelo



- BERT BASE: comparable en tamaño al transformer de OpenAI para comparar el rendimiento
- BERT LARGE: un modelo enorme que logró los resultados de vanguardia

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

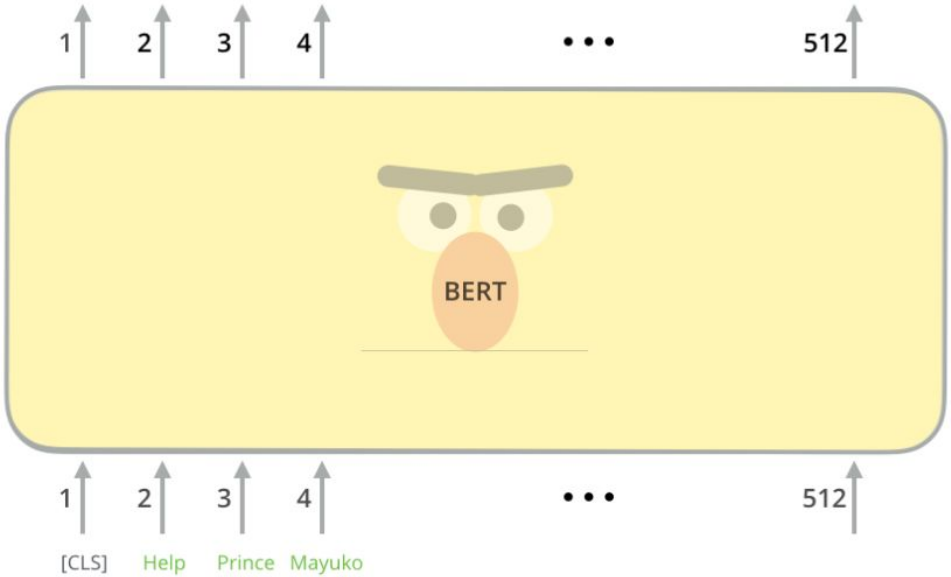
Entradas del modelo



Representación de entrada BERT. Los embeddings de entrada son la suma de los embeddings de tokens, los embeddings de segmentación y los embeddings de posición.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

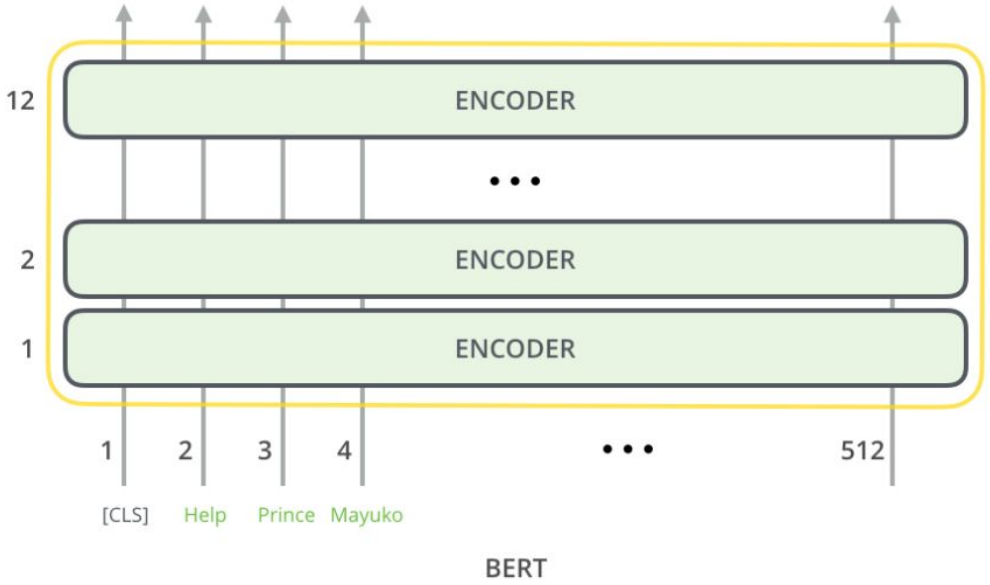
Entradas del modelo



- El primer token de entrada se proporciona con un token [CLS] especial. CLS significa Clasificación.
- La longitud máxima de las secuencias es establecida en 512

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Entradas del modelo

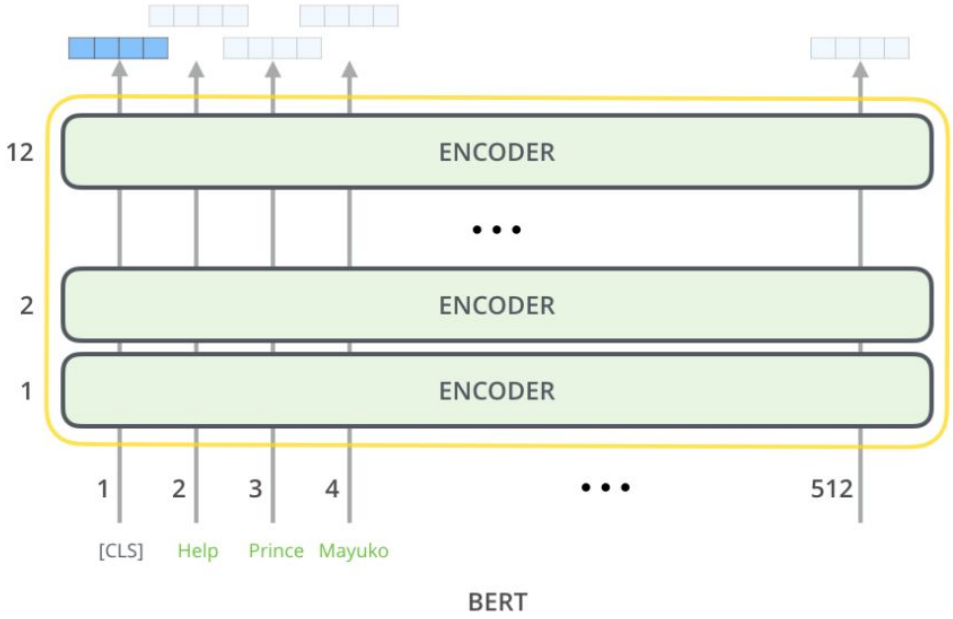


- Al igual que el encoder del transformer, BERT toma una secuencia de palabras como entrada que sigue fluyendo hacia arriba en la pila o bloques.
- Cada capa aplica self-attention y pasa sus resultados a través de una red FFN y luego se los pasa al siguiente encoder.
- En términos de arquitectura, ha sido idéntica a la del Transformer hasta este punto (aparte del tamaño).

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Salidas del modelo

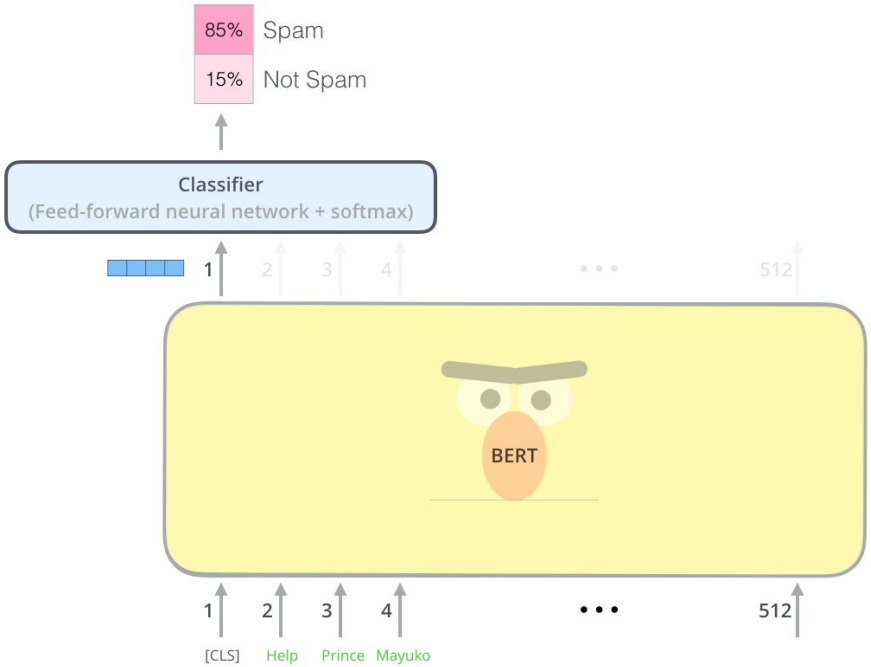


- Cada posición genera un vector de tamaño *hidden_size* (768 en BERT Base).
- Para el ejemplo de clasificación de oraciones, se enfoca en la salida de solo la primera posición (a la que se le pasa el token [CLS] especial).

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Salidas del modelo



- Ese vector ahora se puede usar como entrada para un clasificador de cualquier elección.
- En el paper original logran excelentes resultados simplemente usando una red neuronal de una sola capa como clasificador.

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

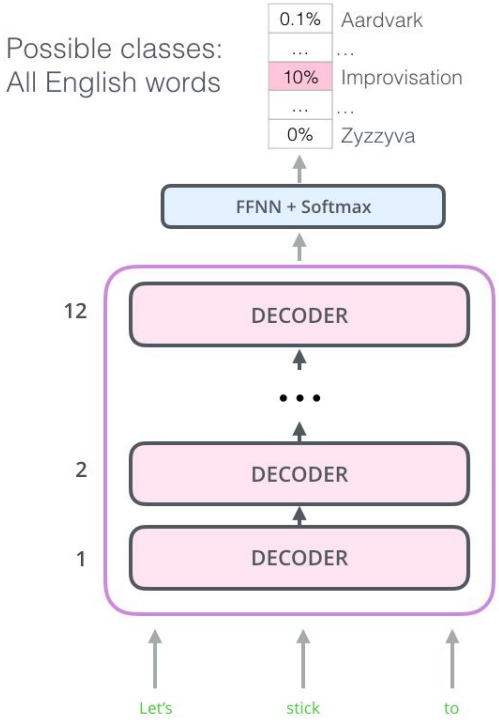
Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Decoders del transformer OpenAI

- El modelo tiene doce capas de decoders apiladas.
- Dado que no hay encoder en esta configuración, estas capas decoders no tendrían la subcapa de atención de encoder-decoder que tienen las capas de decoder del transformer estándar.
- Sin embargo, todavía tendría la capa self-attention (enmascarada para que no alcance su punto máximo en tokens futuros).

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

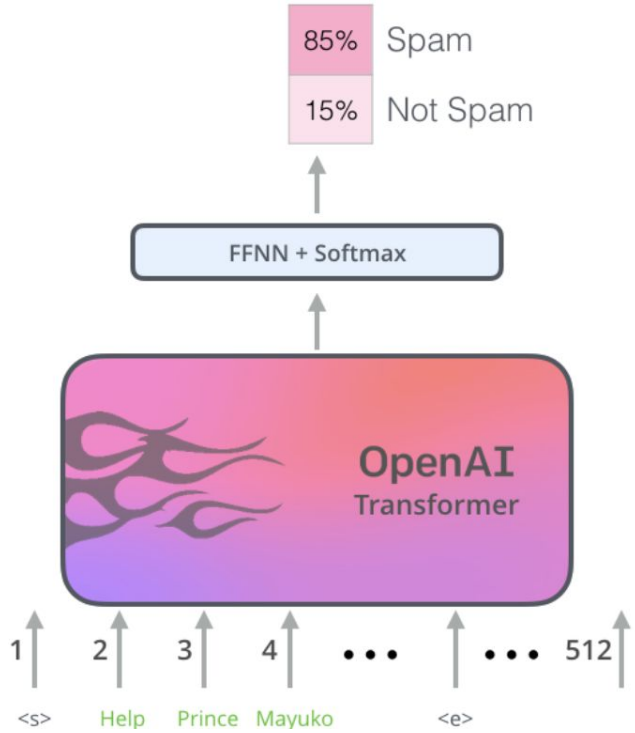
Decoders del transformer OpenAI



- Con esta estructura, se puede proceder a entrenar el modelo en la misma tarea de modelado de lenguaje: predecir la siguiente palabra utilizando conjuntos de datos masivos (sin etiquetar).
- OpenAi transformer fue entrenado con 7,000 libros.
- Los libros son excelentes para este tipo de tareas, ya que permiten que el modelo aprenda a asociar información relacionada, incluso si están separados por mucho texto, algo que no se obtiene, por ejemplo, cuando se entrena con tweets o artículos.

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

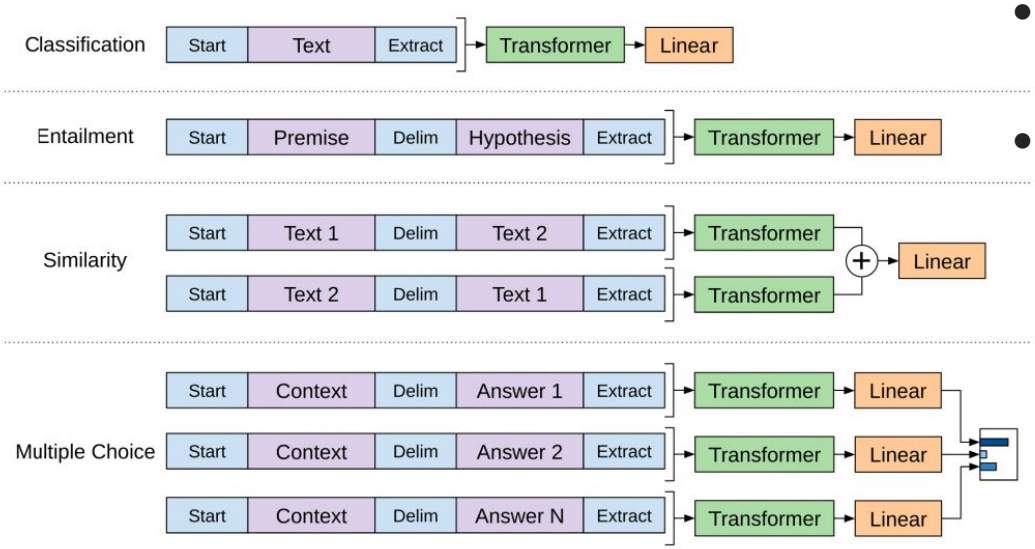
Transfer Learning to Downstream Tasks



- Ahora que el transformer OpenAI está pre-entrenado y sus capas se han ajustado para manejar razonablemente el lenguaje, se puede comenzar a usarlo para downstream tasks (tareas posteriores).
- En cuanto a la clasificación de oraciones (clasificar un mensaje de correo electrónico como "spam" o "no spam"):

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Transfer Learning to Downstream Tasks



- En el paper de OpenAI se describe una serie de transformers de entrada, para manejar las entradas para diferentes tipos de tareas.
- La siguiente imagen del paper muestra las estructuras de los modelos y las transformaciones de entrada para llevar a cabo diferentes tareas.

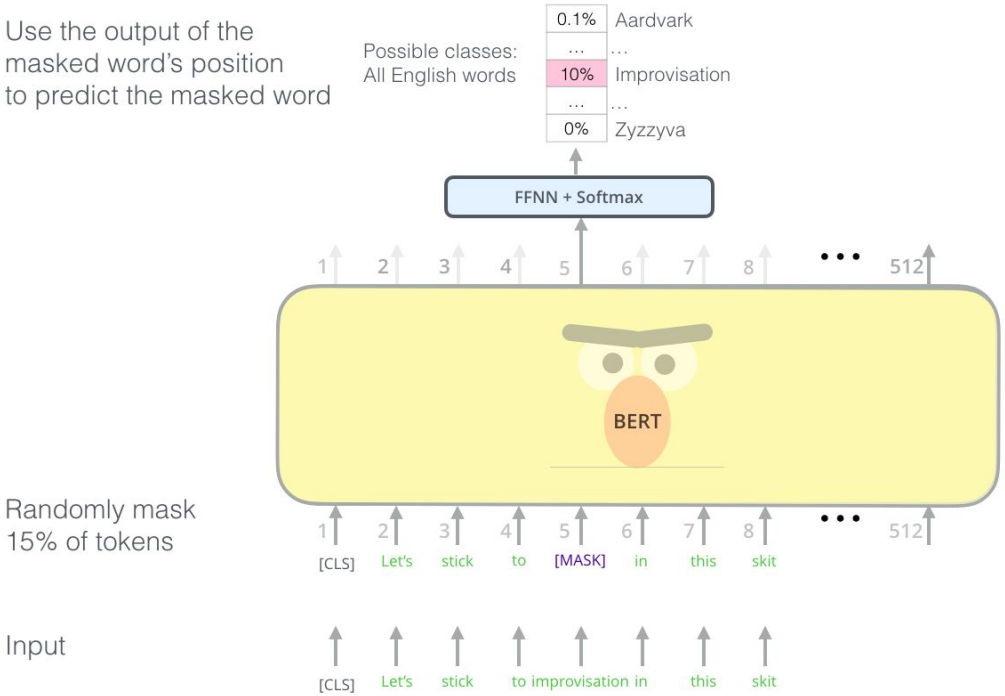
Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

BERT

- El transformer openAI brinda un modelo pre-entrenado ajustable con precisión basado en el transformador original.
- Pero le falta algo en esta transición de LSTM a Transformers.
- El modelo de lenguaje de ELMo es bidireccional, pero el transformer openAI solo entrena un modelo de lenguaje hacia adelante.
- ¿Se puede construir un modelo basado en transformers cuyo modelo de lenguaje mire tanto hacia adelante como hacia atrás?

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

BERT



- Encontrar la tarea correcta para entrenar una pila de encoders de Transformers es un obstáculo complejo que BERT resuelve mediante la adopción de un concepto de "masked language model".
- Más allá de enmascarar el 15% de la entrada, BERT también mezcla un poco las cosas para mejorar la forma en que el modelo se ajusta más tarde.
- A veces reemplaza aleatoriamente una palabra con otra palabra y le pide al modelo que prediga la palabra correcta en esa posición.

Jay Alammar. 2021. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT: Resultados

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT: Resultados

- **MNLI**. Multi-Genre Natural Language Inference es una tarea de clasificación de vinculaciones a gran escala de colaboración abierta. Dado un par de oraciones, el objetivo es predecir si la segunda oración es una vinculación, una contradicción o neutral con respecto a la primera.
- **QQP**. Quora Question Pairs es una tarea de clasificación binaria donde el objetivo es determinar si dos preguntas formuladas en Quora son semánticamente equivalentes.
- **QNLI**. Question Natural Language Inference es una versión del conjunto de datos de respuesta a preguntas de Stanford que se ha convertido en una tarea de clasificación binaria. Los ejemplos positivos son pares (pregunta, oración) que contienen la respuesta correcta, y los ejemplos negativos son (pregunta, oración) del mismo párrafo que no contiene la respuesta.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT: Resultados

- **SST-2.** Stanford Sentiment Treebank es una tarea de clasificación binaria de una sola oración que consta de oraciones extraídas de reseñas de películas con anotaciones humanas de su sentimiento (positivo o negativo).
- **CoLA**. El Corpus of Linguistic Acceptability es una tarea de clasificación binaria de una sola oración, donde el objetivo es predecir si una oración en inglés es lingüísticamente "aceptable" o no.
- **STS-B**. The Semantic Textual Similarity Benchmark es una colección de pares de oraciones extraídas de titulares de noticias y otras fuentes. Se anotaron con una puntuación del 1 al 5 que indica cuán similares son las dos oraciones en términos de significado semántico.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

BERT: Resultados

- **MRPC**. Microsoft Research Paraphrase Corpus consiste en pares de oraciones extraídos automáticamente de fuentes de noticias en línea, con anotaciones humanas para determinar si las oraciones en el par son semánticamente equivalentes.
- **RTE**. Recognizing Textual Entailment es una tarea de vinculación binaria similar a MNLI, pero con muchos menos datos de entrenamiento.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Tokenización

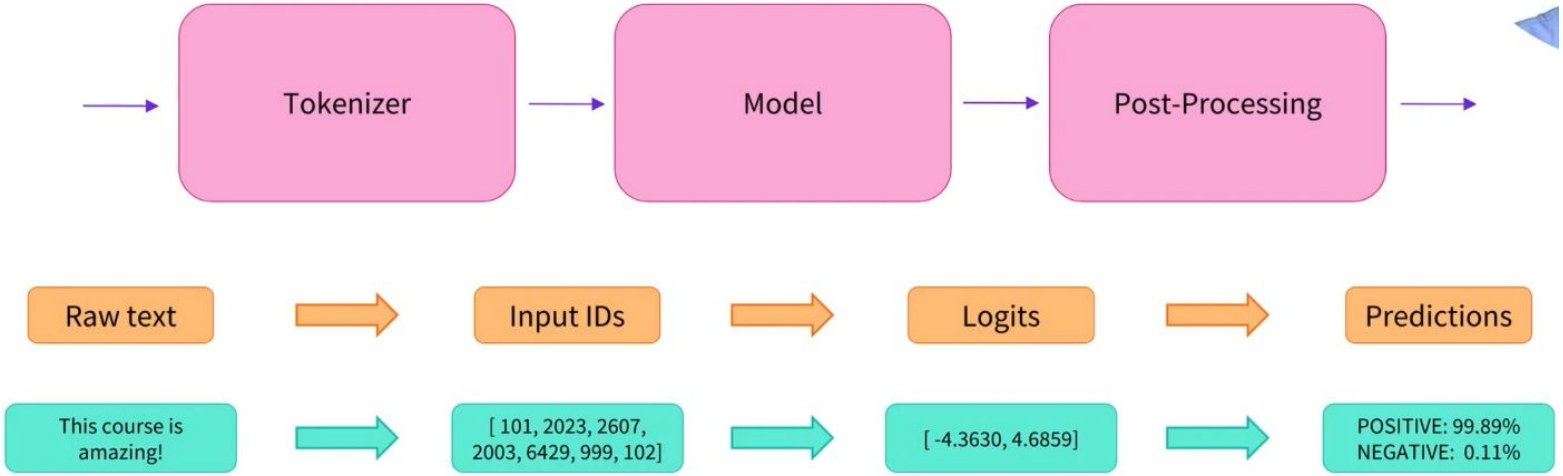
Tokenización

El objetivo del proceso de tokenización es construir un vocabulario basado en los tokens más frecuentes (que es un número predefinido) en todo el conjunto de datos.

Los pasos son los siguientes:

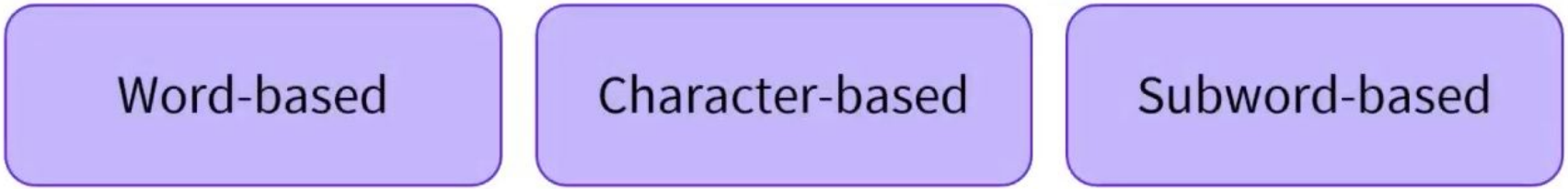
1. Dividir el conjunto de datos en tokens.
2. Contar la cantidad de tokens únicos que aparecieron (frecuencia).
3. Elegir los tokens que aparecen al menos *n* veces.

Tokenización



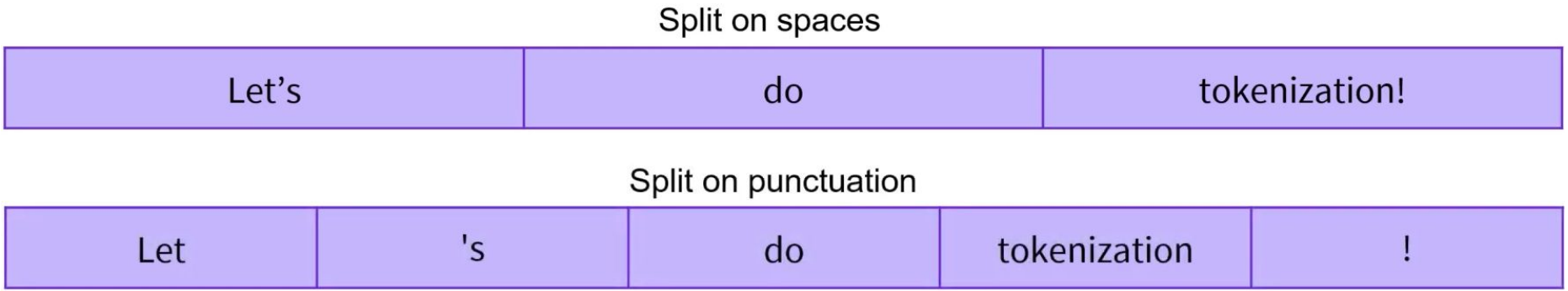
[Hugging Face. Course. Tokenization]

Tokenización



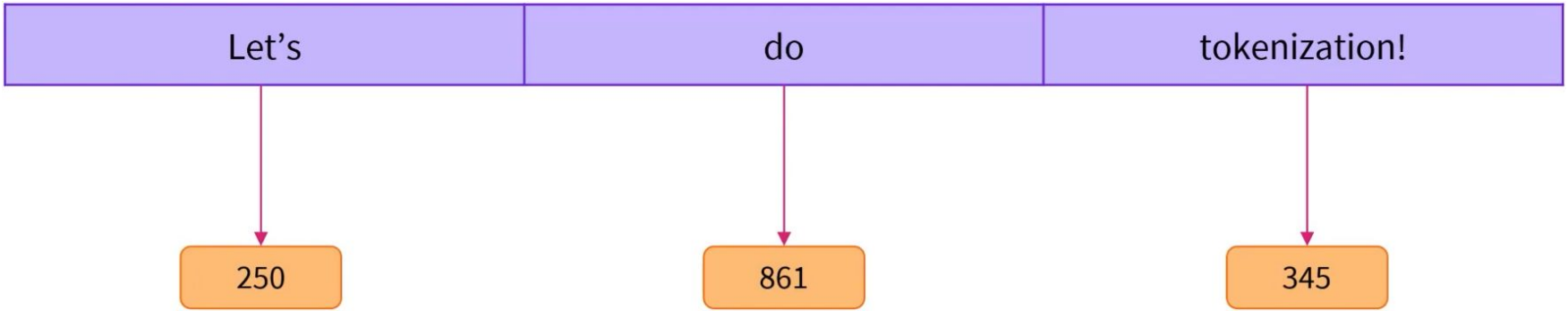
Hugging Face. Course. Tokenization

Tokenización: Basada en palabras



Hugging Face. Course. Tokenization

Tokenización: Basada en palabras



Cada palabra (token) tiene su propio ID

Hugging Face. Course. Tokenization

Tokenización: Basada en palabras

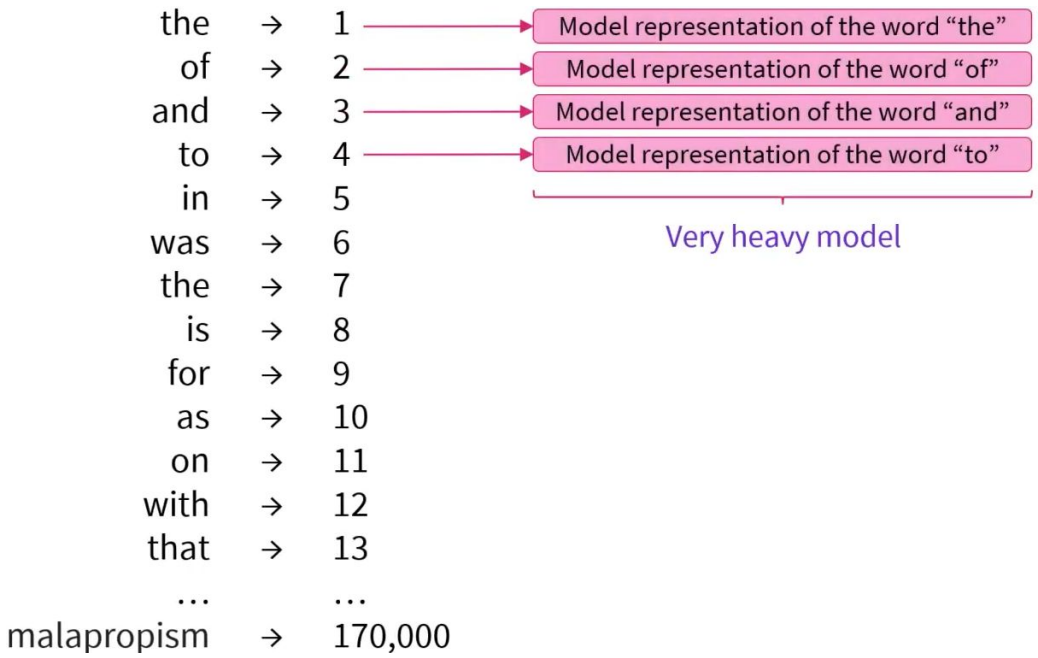
the	→	1
of	→	2
and	→	3
to	→	4
in	→	5
was	→	6
the	→	7
is	→	8
for	→	9
as	→	10
on	→	11
with	→	12
that	→	13
dog	→	14
dogs	→	15

the	→	1
of	→	2
and	→	3
to	→	4
in	→	5
was	→	6
the	→	7
is	→	8
for	→	9
as	→	10
on	→	11
with	→	12
that	→	13
…		…
malapropism	→	170,000

- Palabras similares tienen diferentes significados
- El vocabulario puede tender a ser muy largo

Hugging Face. Course. Tokenization

Tokenización: Basada en palabras



- Los vocabularios grandes resultan en modelos muy pesados

Hugging Face. Course. Tokenization

Tokenización: Basada en palabras

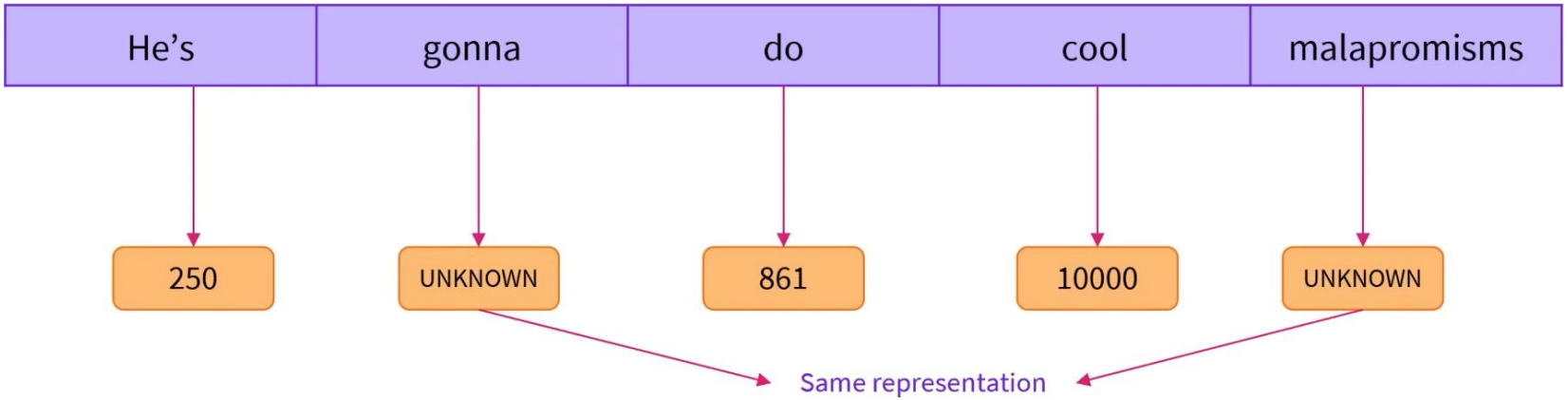
the	→	1
of	→	2
and	→	3
to	→	4
in	→	5
was	→	6
the	→	7
is	→	8
for	→	9
as	→	10
on	→	11
with	→	12
that	→	13
…		…
hug	→	10,000

- Se puede limitar el vocabulario (por ejemplo a 10,000 palabras)

Hugging Face. Course. Tokenization

Tokenización: Basada en palabras

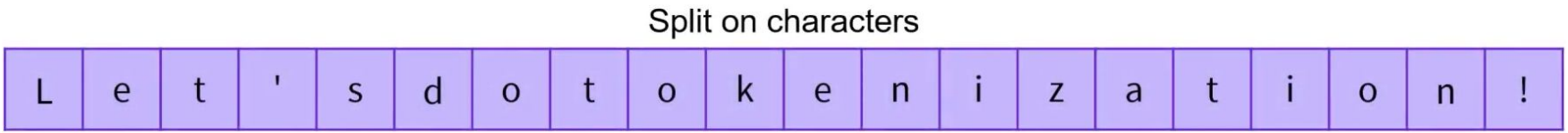
- Sin embargo las palabras fuera del vocabulario (OOV: out of the vocabulary) marcadas como desconocidas (UNK) implica pérdida de información



Hugging Face. Course. Tokenization

Tokenización: Basada en caracteres

- Dividir el texto en caracteres



Hugging Face. Course. Tokenization

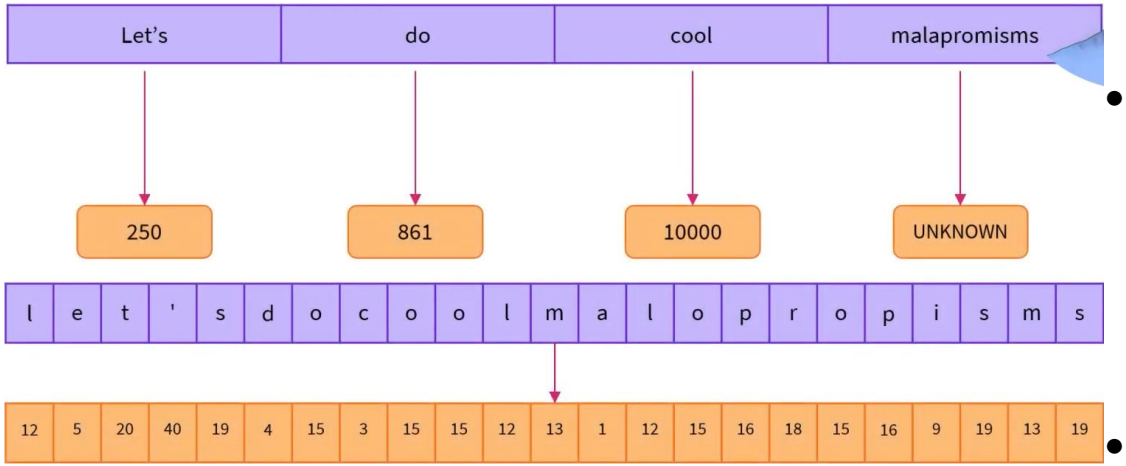
Tokenización: Basada en caracteres

a	→	1	the	→	1
b	→	2	of	→	2
c	→	3	and	→	3
d	→	4	to	→	4
e	→	5	in	→	5
f	→	6	was	→	6
g	→	7	the	→	7
…	→	…	is	→	8
1	→	27	for	→	9
2	→	28	as	→	10
3	→	29	on	→	11
…	→	…	with	→	12
!	→	37	that	→	13
…		…	…		…
à	→	256	malapropism	→	170,000

- El vocabulario de tokenizar con caracteres (izquierda) es mucho más reducido que la tokenización basada en palabras (derecha).

Hugging Face. Course. Tokenization

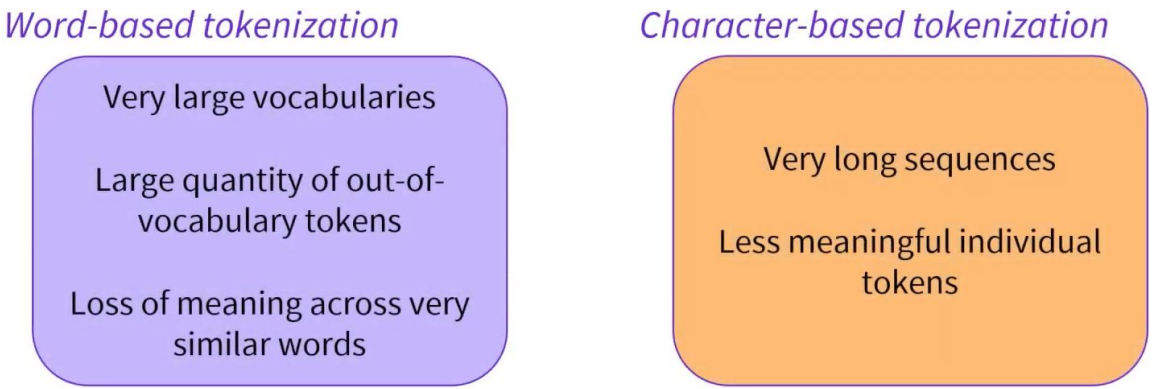
Tokenización: Basada en caracteres



- Hay muchos menos tokens fuera del vocabulario (UNK), ya que cada palabra se puede construir a partir de caracteres.
- Otra cosa a considerar es que ahora se tiene una gran cantidad de tokens para ser procesados por el modelo: mientras que una palabra sería solo un token con un tokenizador basado en palabras, puede convertirse fácilmente en 10 o más tokens cuando se convierte en caracteres.
- Para obtener lo mejor de ambos mundos, se puede usar una tercera técnica que combina los dos enfoques: **tokenización de subpalabras**.

Hugging Face. Course. Tokenization

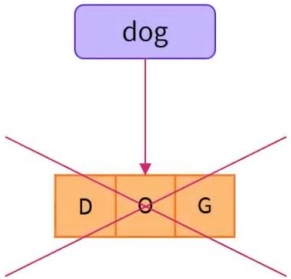
Tokenización: Subpalabras



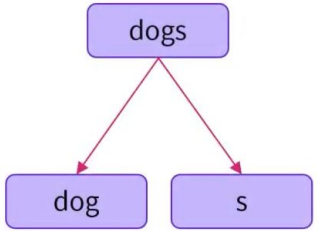
Hugging Face. Course. Tokenization

Tokenización: Subpalabras

Frequently used words should not be split into smaller subwords



Rare words should be decomposed into meaningful subwords.

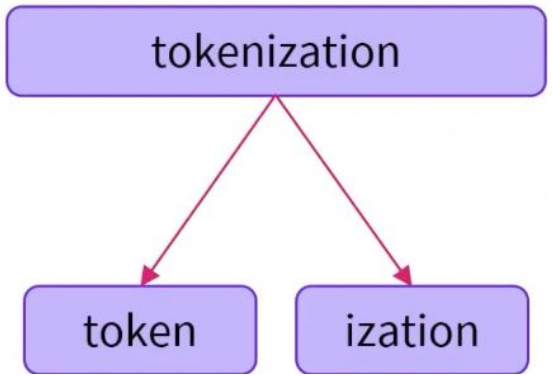


- Los algoritmos de tokenización de subpalabras se basan en el principio de que las palabras de uso frecuente no deben dividirse en subpalabras más pequeñas, pero las palabras raras deben descomponerse en subpalabras significativas.

Hugging Face. Course. Tokenization

Tokenización: Subpalabras

Rare words should be decomposed into meaningful subwords.

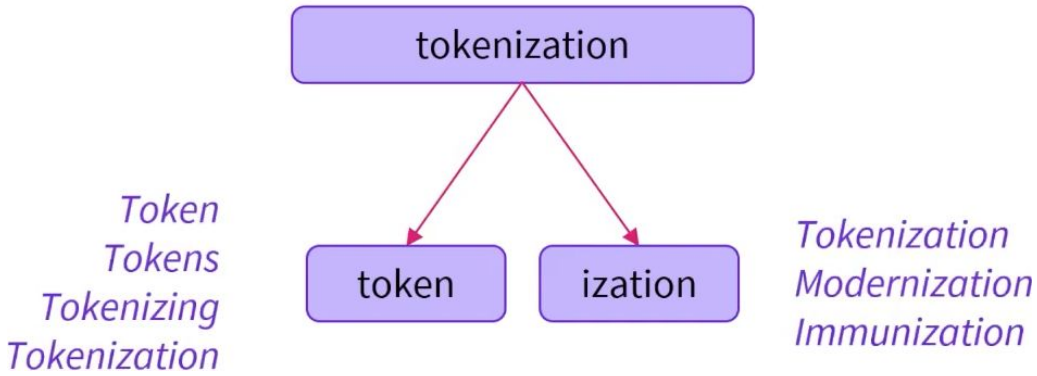


- Los algoritmos de tokenización de subpalabras se basan en el principio de que las palabras de uso frecuente no deben dividirse en subpalabras más pequeñas, pero las palabras raras deben descomponerse en subpalabras significativas.

Hugging Face. Course. Tokenization

Tokenización: Subpalabras

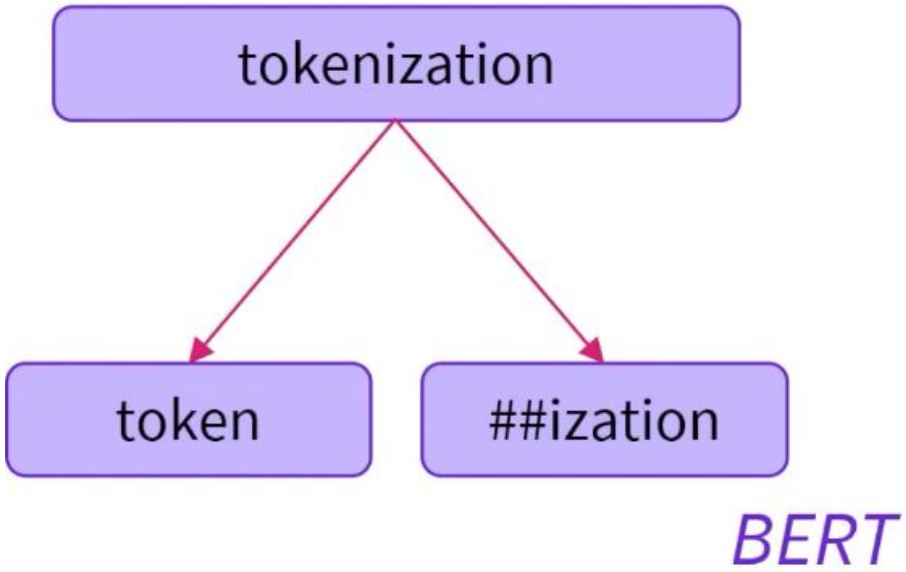
Rare words should be decomposed into meaningful subwords.



- Las subpalabras ayudan a identificar situaciones sintácticas o semánticas similares en el texto

Hugging Face. Course. Tokenization

Tokenización: Subpalabras



- Los algoritmos de tokenización de subpalabras pueden identificar el inicio de tokens de palabras

Hugging Face. Course. Tokenization

Tokenización: Subpalabras

- La mayoría de los modelos que obtienen resultados de vanguardia en inglés hoy en día utilizan algún tipo de algoritmo de tokenización de subpalabras.

Hugging Face. Course. Tokenization