Modelos de Lenguaje

M. en C. Luis Norberto Zúñiga Morales

5 de abril de 2022

Contenido

Modelo de Bolsa de Palabras

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

Figura: Ejemplo de una aplicación de bolsa de palabras.

Modelo de Bolsa de Palabras

- Solo importa la presencia o ausencia de la palabra.
- En consecuencia, no importa su posición.
- Por como se modelan, resulta en un vector disperso.

Modelo de Bolsa de Palabras

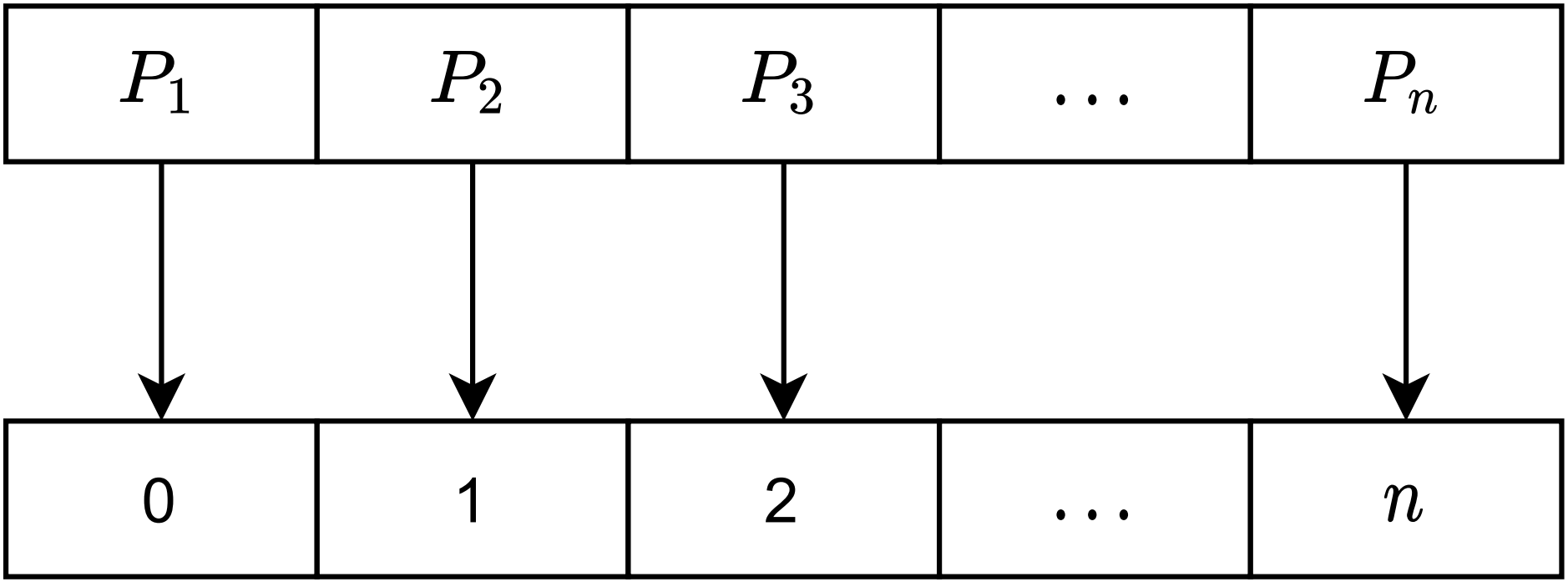

Figura: Idea general de la Bolsa de Palabras. El vector de arriba representa todas las palabras únicas que conforman la colección de documentos. Cada documento se modela con 1 si se encuentra la palabra o 0 si no.

Modelo de Bolsa de Palabras

Ejercicio

Sean los documentos:

- Documento 1: El sol sale por la mañana.
- Documento 2: No ha salido el sol en todo el día.
- Documento 3: Sol, mañana no vas.

Determinar la bolsa de palabras.

N-gramas

- Una alternativa a la bolsa de palabras, bajo la misma idea, pero que considera la posición y relación entre las palabras.
- Se «unen» palabras de *n* en *n*.
- Se puede añadir una distancia o radio de acción.

N-gramas

Para considerar el contexto de una palabra en una oración, adoptamos la idea de Njolstad et al. [1], donde utilizan **Términos de Coocurrencia**[1] (TCO). Se establece una vecindad de radio r y se forman grupos de n palabras.

Ejemplo (términos de coocurrencia con $n = 2$ y $r = 3$)

No me parece que este sea el destino de Leonardo, Federico. La pregunta es esta: ¿el destino llega sin nuestra voluntad? ¿O somos nosotros quienes lo provocamos?

↓

[(No, me),(No, parece), (No, que), (me, parece), (me, que), (me, este),...]

[1] La utilización conjunta de dos o más palabras en un documento.

N-gramas

Sin embargo, utilizamos la puntuación que designa el fin de una idea para delimitar el alcance de los TCO:

.?!:;

Ejemplo (términos de coocurrencia con $n = 2$ y $r = 3$ con delimitadores)

No me parece que este sea el destino de Leonardo, Federico. La pregunta es esta: ¿el destino llega sin nuestra voluntad? ¿O somos nosotros quienes lo provocamos?

↓

[...,(Federico, .), (La, pregunta),...]

Si $n = 0$ se extrae palabra por palabra. Si $r = 1$ y $n > 0$ se presenta el caso de n-gramas[2].

[2] Un n-grama es una subsecuencia de n elementos de una secuencia dada.

Word Embeddings

- La idea principal al trabajar con texto es encontrar una representación matemática para cada palabra.
- Los *word embbedings* consisten en representar palabras por medio de vectores $\boldsymbol{v} \in \mathbb{R}^n$.

$$f : W \rightarrow \mathbb{R}^n$$

- El método más usado para generar *word-embeddings* es word2vec [2] en su modalidad *Skip-grams*.
- Recientemente BERT [3] ha superado a word2vec como modelo para obtener *word embeddings*.

Un texto es una secuencia de T palabras, donde w_i indica la i-ésima palabra del texto a representar:

w_1	$\cdots$	w_{i-1}	w_i	w_{i+1}	$\cdots$	w_T

Para w_i, su contexto se obtiene al observar las palabras que tiene adelante y atrás:

$$\underbrace{w_{i-M}, \ldots, w_{i-1}}_{\text{contexto anterior}}, \overbrace{w_i}^{\text{centro}}, \underbrace{w_{i+1}, \ldots, w_{i+M}}_{\text{contexto posterior}}$$

donde M es la mitad del tamaño de la ventana utilizada para barrer el texto.

Extracción de Características con word2vec

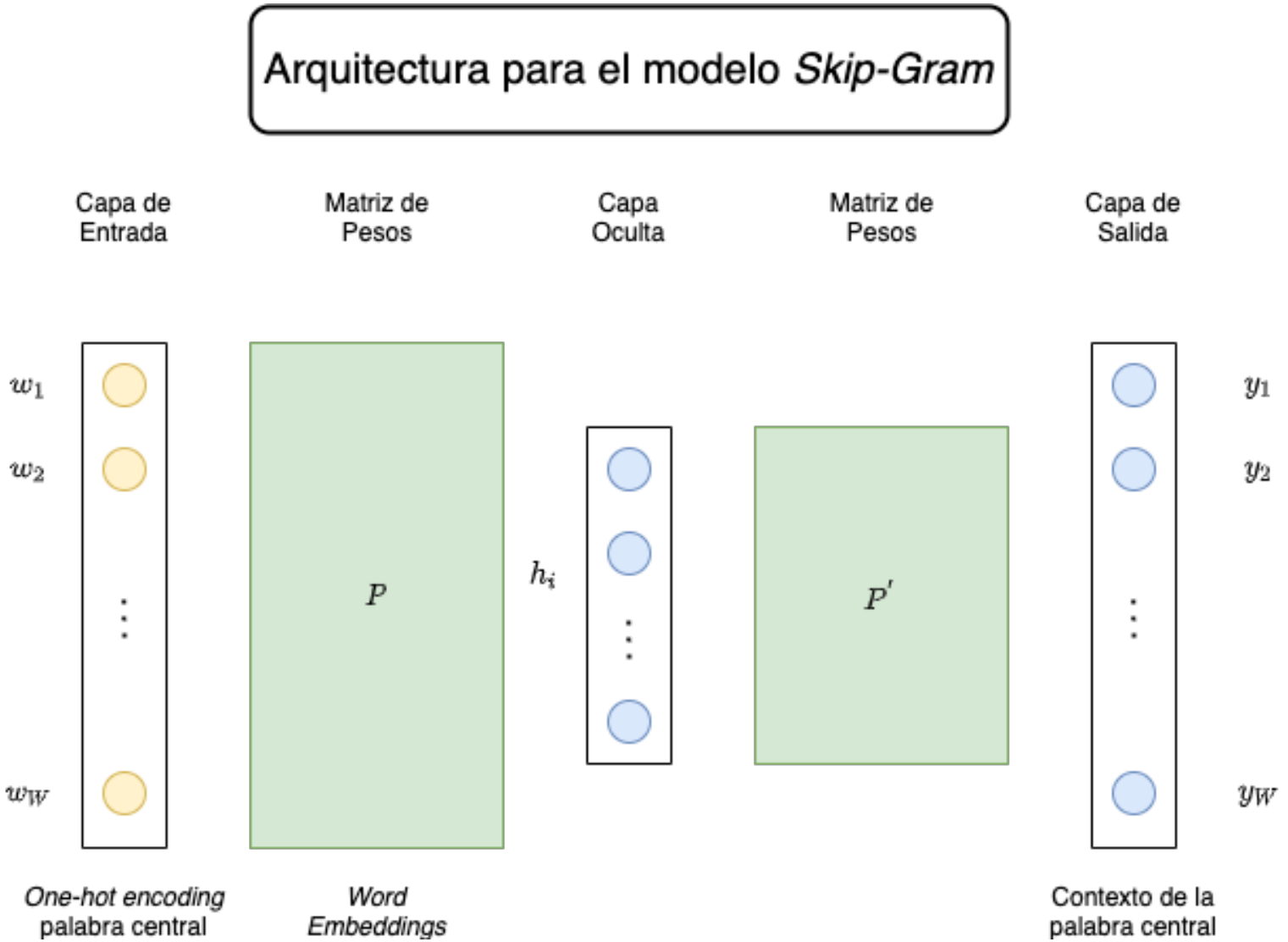

Figura: Arquitectura de la red neuronal que emplea word2vec para extraer *word embeddings*. Fuente: Realización propia.

BERT

- Devlin et al. [3] propusieron un nuevo modelo para generar *word embeddings*, BERT (*Bidirectional Encoder Representations from Transformers*).
- BERT logró superar a diversos modelos de lenguaje en múltiples tareas de procesamiento de lenguaje natural [3].
- BERT se basa en la arquitectura del *Transformer* [4], específicamente en su codificador (*encoder*).

BERT

- Considera entrenamiento bidireccional simultaneo y busca predecir el vocabulario considerando únicamente su contexto.
- Utiliza la idea del pre-entrenamiento , i.e., se puede ajustar a múltiples tareas de procesamiento de lenguaje natural.
- El modelo se pre-entrena por medio de dos tareas: predicción de palabras y predicción de enunciados .

Extracción de Características con BERT

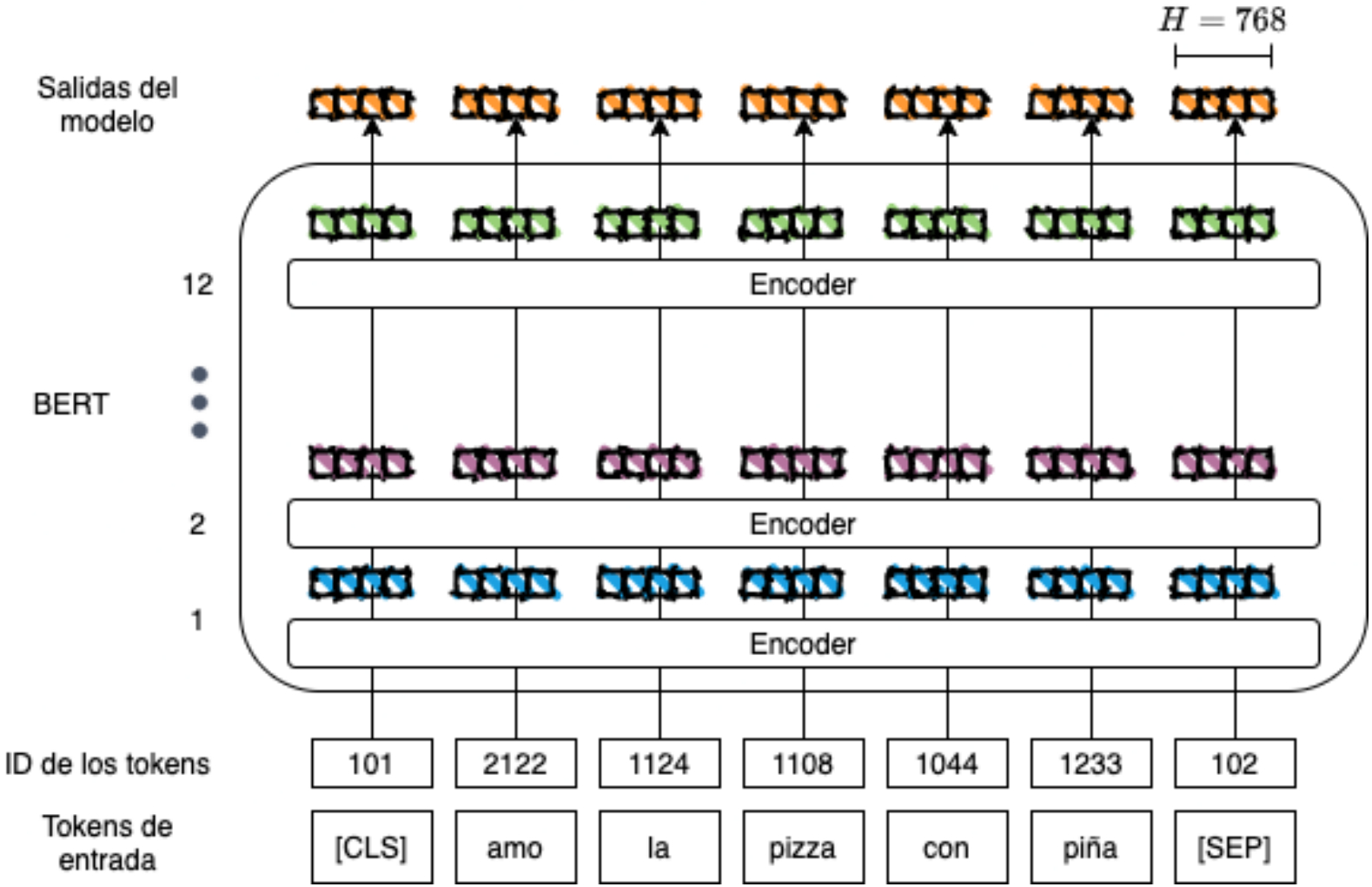

Figura: Esquema general de BERT [3] para extraer *word embeddings*.
Fuente: Realización propia.

Modelos para Representar Texto

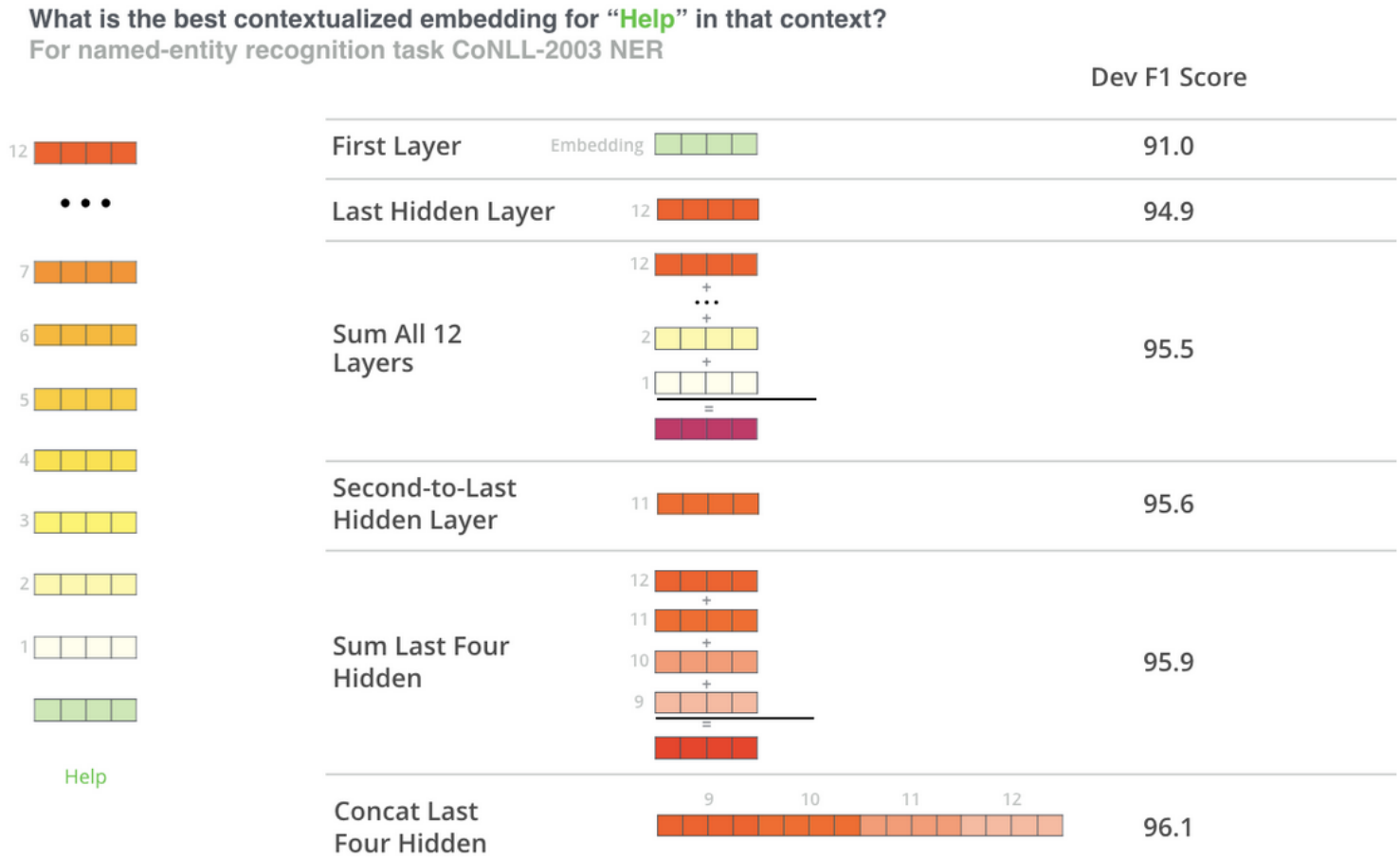

Figura: Opciones y resultados para obtener *word embeddings* con Bert.
Fuente: jalammar.github.io

Detalles adicionales de BERT

- El tamaño máximo de tokens es 512.
- Utilizan WordPiece tokenization [5].
- Batch: 16, 32
- Epochs: 2, 3, 4

.

Implementaciones de BERT

- Existen diversas implementaciones de BERT en distintos idiomas y de distinto tamaño.
- Es posible entrenar modelos y utilizar dichas implementaciones por medio de Tensorflow y Huggingface.
- Existe una implementación de BERT en español, BETO [6] disponible en las plataformas anteriores.

Tareas de PLN

- MNLI, Multi-Genre Natural Language Inference
- QQP, Quora Question Pairs
- QNLI, Question Natural Language Inference
- SST-2 The Stanford Sentiment Treebank
- CoLA, The Corpus of Linguistic Acceptability
- STS-B The Semantic Textual Similarity Benchmark

Tareas de PLN

- MRPC, Microsoft Research Paraphrase Corpus
- RTE Recognizing Textual Entailment

Referencias I

[1] P. C. S. Njølstad, L. S. Høysæther, and J. A. Gulla. Optimizing supervised sentiment lexicon acquisition: Selecting co-occurring terms to annotate for sentiment analysis of financial news. 2014.

[2] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. January 2013.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. October 2018.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. June 2017.

Referencias II

[5] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. September 2016.

[6] José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*, 2020.