MAPAS DEL LABERINTO

UNA SERIE DE PHANTOM MAZE — AI & LANGUAGE LAB

ENSAYO 20

Embeddings: el mapa sin cartógrafo

Claudia Marsico · Hernán Inverso

DOI 10.5281/zenodo.23647190

El lenguaje puede convertirse en geometría. Los embeddings muestran hasta dónde puede llegar una máquina guiándose por relaciones



Marc Chagall, Yo y la aldea (1911)

Antístenes, el socrático que se burlaba de Platón, renegaba de las definiciones y esgrimía su propia receta: para saber qué es una cosa hay que situarla entre otras. La idea tiene un aire sorprendentemente moderno. Una palabra está en una red de relaciones con las que tiene semejanzas, oposiciones o asociaciones habituales. La proto-teoría de campos semánticos de Antístenes tiene su eco en los embeddings, que ubican unidades lingüísticas en un espacio matemático según las regularidades que

aprendieron de sus usos. El parecido termina ahí, pero el movimiento es semejante: en lugar de preguntar qué significa algo hay que preguntar dónde queda respecto de lo demás.

Cuando hablamos de tokens nos referimos al texto cortado en piezas discretas de un vocabulario computacional, que no necesariamente corresponden a palabras, letras, morfemas ni unidades de sentido. Pero una ficha por sí sola todavía no le sirve a una red neuronal porque para calcular necesita números. El paso siguiente consiste entonces en transformar cada token en un vector, es decir, una lista ordenada de números que pueden imaginarse como coordenadas. Con dos coordenadas ubicamos un punto en una hoja; con tres, un punto en una habitación. Con tres mil seguimos teniendo un punto, aunque ya no podamos dibujar el espacio en el que vive, manejable computacionalmente.

Marc Chagall, Yo y la aldea (1911)

¿Cómo recibe cada token semejante lista? En un modelo hay una gran tabla aprendida durante el entrenamiento, la matriz de embeddings. Cada ficha del vocabulario tiene asociada una fila de esa tabla. Cuando aparece un token, el sistema toma la fila correspondiente y obtiene así su primera representación numérica. El identificador del token, supongamos que "perro" fuera un único token y tuviera el identificador 1734, no significa nada por sí mismo. Es apenas una dirección para ir a buscar su vector. Ese vector es denso, porque está compuesto por muchos valores numéricos distintos de cero, es aprendido, porque nadie decidió a mano cuáles debían ser sus números, y es distribuido, porque una propiedad relevante no vive en una sola coordenada, como si el número 82 significara "animal" y el 417 "plural". Las regularidades se reparten entre muchas dimensiones y adquieren sentido en conjunto.

El verbo inglés *to embed* significa incrustar, fijar algo dentro de una materia que lo rodea, como una piedra en el cemento. Un *embedding* incrusta una unidad de lenguaje en un espacio poblado por otras. El token que en el ensayo anterior era una ficha separada de todas las demás se convierte ahora en una posición desde la cual pueden calcularse proximidades, diferencias y direcciones. La intuición lingüística que vuelve útil esa geometría venía de mediados del siglo XX. Zellig Harris propuso estudiar las unidades de una lengua observando su distribución, es decir, los contextos en que aparecen. Dos palabras que pueden ocupar lugares semejantes en muchas construcciones tienen algo en común detectable por análisis distribucional sin definiciones. John Firth condensó pocos años después la intuición en la frase: "conocerás una palabra por la compañía que mantiene". El uso deja huellas. "Perro" aparece una y otra vez en entornos donde encontramos ladrar, correa, paseo, mascota, veterinario; "hipoteca", cerca de banco, cuota, interés, vivienda, préstamo. Si reunimos cantidades suficientes de texto, esas regularidades empiezan a formar una especie de firma estadística.

En 2013, Tomas Mikolov y su grupo presentaron word2vec, que encontró una manera eficaz de comprimir esas huellas en vectores. En una variante, la tarea de entrenamiento utilizaba una palabra para predecir cuáles suelen aparecer cerca; en la otra, reconstruía una palabra a partir de las que la rodean. Para mejorar sus predicciones, sin diccionario ni clasificación previa de términos, el

entrenamiento modificaba una y otra vez los números asociados a cada palabra. Al final, las palabras utilizadas en contextos semejantes tendían a quedar representadas por vectores también semejantes.

Pero ¿qué quiere decir que dos listas de cientos de números sean "semejantes"? Imaginemos cada vector como una flecha que parte del origen de ese espacio enorme. Si dos flechas apuntan aproximadamente hacia el mismo lado, tienen una gran similitud de coseno; si forman un ángulo mayor, esa similitud disminuye. Así nace el mapa. Palabras usadas de maneras parecidas tienden a ocupar regiones próximas. "Perro" y "gato" pueden quedar cerca porque aparecen en contextos semejantes, pero lo mismo puede ocurrir con términos opuestos. "Frío" y "caliente" comparten muchísimas compañías asociadas con temperatura, agua, clima, sensación, invierno o verano. El espacio registra primero regularidades de uso y a partir de la combinación de muchas de ellas puede distinguir clases de relaciones diferentes.

La sorpresa de word2vec fue que, además de formar vecindarios, algunas relaciones repetidas por la lengua aparecían como desplazamientos aproximadamente regulares. Ciertos plurales respecto de sus singulares, algunos tiempos verbales, capitales respecto de países o formas femeninas respecto de masculinas podían encontrarse siguiendo direcciones semejantes. De ahí nació la operación que hizo célebre el modelo: tomar el vector de "rey", restarle el de "hombre", sumarle el de "mujer" y terminar cerca de "reina". Las analogías no funcionan siempre, dependen mucho del corpus y del modo de entrenamiento y no prueban que exista un álgebra universal escondida detrás de las palabras, pero una relación que nadie había programado explícitamente adquirió una forma geométrica. La estadística del uso produjo direcciones.

En el siglo XVII, Leibniz soñó con una *characteristica universalis*, un alfabeto de los conceptos en el que cada idea tuviera asignado un signo, de modo que razonar fuera calcular. Ante cualquier disputa, en lugar de discutir bastaría con sentarse y decir *calculemus*, calculemos, y hacer las cuentas hasta dar con la razón. Los *embeddings* rozan ese sueño extravagante hecho de conceptos convertidos en vectores, analogías expresables como desplazamientos aritméticos, aunque han perdido por el camino la taxonomía racional que Leibniz esperaba. También ponen de cabeza el ideal de lengua adánica donde el nombre refleja la esencia de la cosa. El embedding se desentiende de qué es un perro y solo le interesa con qué aparece y en qué contextos se mueve. Donde la lengua adánica tenía esencia, el espacio vectorial encuentra vecindad.

La geometrización del lenguaje se asomó en el análisis semántico latente, desarrollado alrededor de 1990. El procedimiento partía de una enorme tabla que registraba qué términos aparecían en qué documentos y aplicaba técnicas algebraicas para comprimirla. Esa compresión hacía emerger proximidades que no eran evidentes en la tabla original, pero word2vec llevó el principio a una escala y una precisión nuevas al aprender representaciones mediante predicción local. GloVe, presentado por investigadores de Stanford en 2014, por su parte, explotó estadísticas globales de coocurrencia. Aunque los métodos diferían, compartían la limitación de que cada palabra recibía un único vector fijo. "Banco" tenía que vivir siempre en el mismo punto, ya fuera el asiento de una plaza o la institución donde se deposita dinero. El vector terminaba siendo una especie de promedio de sus usos.

¿Y si una palabra no tuviera una posición única? ELMo exploró en 2018 la idea de construir representaciones que dependieran de la oración completa. Ese mismo año BERT, apoyado ya en la arquitectura transformer, generalizó y popularizó el cambio. Desde entonces conviene distinguir dos cosas. Al ingresar al modelo, un token recibe un embedding inicial relativamente estable, tomado de la tabla que describimos antes. Pero ése es sólo el punto de partida, ya que a medida que atraviesa las capas, su vector cambia por efecto del contexto. Así, el token correspondiente a "banco" parte del mismo embedding léxico básico, aunque la representación de entrada incorpora también información sobre su posición y termina ocupando posiciones distintas después de que la arquitectura incorpora el contexto. A esos vectores posteriores solemos llamarlos representaciones contextuales o, en un sentido más amplio, embeddings contextuales. En lugar de un único punto para "banco" hay una posición construida para esa aparición concreta. La atención es uno de los mecanismos centrales que producen esa transformación y de ese modo el mapa, en lugar de un atlas impreso que asigna a cada palabra una ciudad fija, resulta un mapa que se redibuja mientras se lee la frase.

La misma idea geométrica se expandió luego más allá de las palabras. Sentence-BERT, en 2019, produjo vectores especialmente útiles para representar frases y párrafos completos, de modo que dos textos pudieran compararse por su cercanía semántica. CLIP, en 2021, llevó el principio al territorio de imágenes y textos para que ambos tipos de objeto pudieran ocupar regiones compatibles de un mismo espacio. Una fotografía de un perro y una descripción apropiada del perro podían quedar cerca aun siendo cosas físicamente incomparables, haciendo convivir modalidades distintas dentro de una misma geometría aprendida.

¿Para qué sirve esto? Una búsqueda tradicional se apoya en coincidencias de términos, mientras una búsqueda vectorial puede convertir preguntas o documentos en vectores y recuperar los que quedaron más cerca, aunque no utilicen el mismo vocabulario. Preguntar por "cómo bajar el costo de calefaccionar una casa" puede recuperar un pasaje sobre "mejorar el aislamiento térmico", aunque las frases no compartan términos. Para hacer ese tipo de búsquedas a gran escala se usan bases de datos vectoriales, diseñadas para almacenar y encontrar rápidamente vecinos dentro de colecciones inmensas de vectores. Esa búsqueda es también una pieza central de muchos sistemas RAG, sigla de Retrieval-Augmented Generation, generación aumentada por recuperación. Antes de recibir preguntas, el sistema prepara una colección de textos: los divide en fragmentos, calcula un vector para cada uno y los deja indexados. Cuando llega una consulta, calcula también su vector y busca cuáles de esos fragmentos quedaron más cerca. Luego coloca los pasajes recuperados junto a la pregunta y se los entrega al modelo de lenguaje. El embedding ayuda a decidir qué información conviene poner delante del modelo antes de que responda. La misma lógica sirve para recomendar contenido parecido a lo que alguien ya consumió, agrupar documentos por afinidad, detectar casos atípicos, ordenar grandes archivos o relacionar textos e imágenes. Lo que empezó como una manera de representar palabras terminó convertido en un procedimiento general para transformar objetos complejos en puntos comparables y medir entre ellos distintas formas de semejanza.

Volvamos al punto más extraño. Ningún cartógrafo dibujó este mapa ni decidió de antemano qué debía significar cada eje. Latitud y longitud son una decisión humana, con un meridiano cero que

quedó en un observatorio de Londres como una bandera de poder naval en el s. XIX. En un espacio de embeddings nadie comenzó diciendo que cierta dimensión representaría el género, otra la temporalidad, otra la autoridad y otra la relación entre un animal y su sonido. El entrenamiento se enfocó en predecir contextos y ajustó los números hasta hacerlo cada vez mejor. La geometría apareció como resultado. Podemos explorarla entrenando sondas que detecten información gramatical o semántica e incluso intervenir sobre algunos patrones, pero el mapa funciona mejor de lo que podemos glosarlo.

Comenzamos con Antístenes abandonando la esencia por la relación, cambiando la pregunta qué es un perro por dónde queda. En eso el embedding es una suerte de hijo suyo. El vínculo filial estalla cuando Antístenes afirma que cada nombre refiere a una cosa existente hasta la paradoja de que el error es imposible: quien dice algo, dice algo que es y quien dice algo que es, dice la verdad. El embedding cortó amarras con el mundo. "Perro" recibe su posición porque la palabra apareció millones de veces cerca de ladrar, de correa y de veterinario. Se comparan palabras por su vecindad y el resultado es una teoría del sentido que funciona sin que el mundo intervenga. Diseñamos una tecnología que puede prescindir de la pregunta por la correspondencia. El mapa representa las huellas que dejaron unos términos al aparecer junto a otros. Cabe recordarlo porque a nosotros sí nos interesan el mundo y las cosas. El embedding muestra hasta dónde puede llegar una inteligencia que se orienta por relaciones sin empezar por una definición esencial y también hace patente el costo de esa libertad, ya que puede aprender a moverse con una precisión asombrosa entre nuestras palabras sin que por eso sus criterios de relevancia coincidan con los nuestros. Así, este mapa puede ser excelente para recorrer el lenguaje y aun así no decirnos dónde están las fronteras que importan en el mundo.

◊

CÓMO CITAR

Marsico, Claudia y Hernán Inverso. 2026. "Embeddings: el mapa sin cartógrafo". *Mapas del laberinto*, ensayo 20. Phantom Maze — AI & Language Lab. https://doi.org/10.5281/zenodo.23647190

Mapas del laberinto es una serie de Phantom Maze, AI & Language Lab. La colección vive en **phantommaze.cloud**; los ensayos aparecen también en phantommazeailabes.substack.com. La suscripción es gratuita.

Algunas de las ideas que enmarcan este texto se desarrollan en *El fantasma sin ego. Un mapa en el laberinto de la inteligencia artificial*, de Hernán Inverso y Claudia Marsico (Phantom Maze Press, 2026; ISBN 978-987-3729-10-2).

PHANTOM MAZE

Leer este ensayo en línea · Todos los ensayos · El libro

Fuentes. Scott Deerwester et al., "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990, pp. 391–407; Jacob Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv*, 11 Oct. 2018, arXiv:1810.04805; J. R. Firth, "A Synopsis of Linguistic Theory, 1930–

1955," *Studies in Linguistic Analysis*, special volume of the Philological Society, Basil Blackwell, 1957, pp. 1–32; Zellig S. Harris, "Distributional Structure," *Word*, vol. 10, nos. 2–3, 1954, pp. 146–62; Gottfried Wilhelm Leibniz, *Dissertatio de arte combinatoria*, Fick and Seubold, 1666; Tomas Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," *arXiv*, 16 Jan. 2013, arXiv:1301.3781; Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig, "Linguistic Regularities in Continuous Space Word Representations," *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, 2013, pp. 746–51; Jeffrey Pennington et al., "GloVe: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2014, pp. 1532–43; Matthew E. Peters et al., "Deep Contextualized Word Representations," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, Association for Computational Linguistics, 2018, pp. 2227–37; Alec Radford et al., "Learning Transferable Visual Models from Natural Language Supervision," *Proceedings of the 38th International Conference on Machine Learning*, edited by Marina Meila and Tong Zhang, vol. 139, Proceedings of Machine Learning Research, 2021, pp. 8748–63; Nils Reimers and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019, pp. 3982–92.