

Teoría y práctica de redes neuronales: taller de modelos de lenguaje para la gestión pública

Mgtr. Alberto Martínez del Pezzo¹

Ushuaia, Tierra del Fuego, Argentina. 7, 8 y 9 de octubre de 2026.

¹ Mgtr. Alberto Martínez del Pezzo es Magíster en Ciencias Sociales por la Universidad Nacional de Quilmes y Licenciado en Ciencia Política por la Universidad de Buenos Aires. Es especialista en Procesamiento de Lenguaje Natural. Se desempeña como docente experto en el Instituto Nacional de Administración Pública (INAP), donde dicta los seminarios Minería de Texto en la Oficina y Administración Pública y Herramientas Informáticas Basadas en IA. Asimismo, es profesor-investigador en la Universidad Nacional de Chilecito.

Teoría y práctica de redes neuronales: taller de modelos de lenguaje para la gestión pública

Resumen

La ponencia aborda la evolución del procesamiento del lenguaje natural desde un problema central: cómo representar la similitud entre palabras y, a la vez, distinguir sus significados según el contexto. A partir de Wittgenstein, se recorre el pasaje desde TF-IDF y Word2Vec hacia Transformer, BERT y GPT, destacando atención, desambiguación y predicción causal. Transformer Explainer permite visualizar estos mecanismos sobre GPT-2. Finalmente, dos experiencias con BETO², mediante Gradio, muestran la predicción de palabras enmascaradas y la extracción de entidades nombradas. La propuesta vincula estos desarrollos con usos introductorios en gestión pública, sin requerir conocimientos de programación.

Palabras clave: SIMILITUD SEMÁNTICA; CONTEXTO; TRANSFORMERS; DESAMBIGUACIÓN; INTELIGENCIA ARTIFICIAL GENERATIVA.

1. Similitud y contexto como problema del significado

Esta ponencia parte de una pregunta que antecede ampliamente a los modelos contemporáneos de inteligencia artificial: ¿cómo podemos establecer que dos palabras o expresiones se parecen en significado y, al mismo tiempo, distinguir cuándo una misma palabra significa cosas distintas según el contexto? La pregunta combina dos fenómenos que para una persona hablante suelen resolverse de manera casi inmediata. Por un lado, reconocemos semejanzas entre expresiones diferentes; por otro, desambiguamos una misma forma lingüística atendiendo a la situación en la que aparece. Para una computadora, en cambio, ambas operaciones requieren construir alguna representación del lenguaje que pueda compararse y modificarse.

La reflexión de Ludwig Wittgenstein ofrece un punto de partida especialmente fértil para formular este problema. En Investigaciones filosóficas sostiene, para una amplia clase de casos, que “el significado de una palabra es su uso en el lenguaje” (Wittgenstein, 1988, 43). En los párrafos dedicados a los llamados parecidos de familia también cuestiona la idea de que toda categoría deba definirse por una esencia única y común a todos sus miembros. No se propone aquí una genealogía directa entre esa filosofía y los algoritmos de procesamiento del lenguaje natural. La utilidad del planteo es otra: desplaza la atención desde la palabra aislada hacia las relaciones que su uso establece con otras expresiones y con situaciones concretas.

Ese desplazamiento permite comprender por qué la similitud y el contexto no son problemas opuestos, sino complementarios. Si dos términos aparecen en usos próximos, es razonable preguntarse si esas regularidades pueden convertirse en una medida de semejanza. Pero, a la vez, una misma palabra puede participar en usos muy diferentes. La lingüística distribucional del siglo XX convirtió parte de esa intuición en un programa empírico: Harris (1954) propuso describir los elementos lingüísticos a partir de sus distribuciones y entornos de aparición, mientras que Firth (1957) destacó la importancia de la compañía que mantienen las palabras. Esta tradición es relevante porque

² La notebook puede ser descargada del siguiente link
https://github.com/amartinezdelpezzo/Taller_BETO_AAEAP2026_MASK_NER_QR_v2

introduce una idea que luego será central en los modelos vectoriales: las regularidades de coocurrencia pueden aportar información sobre relaciones de significado.

Considérese la palabra banco. En “El ciudadano solicitó un crédito en el banco”, términos como crédito, dinero o cuenta orientan hacia una institución financiera. En “El ciudadano se sentó en el banco de la plaza”, palabras como sentarse, plaza o parque orientan hacia un asiento. El término es idéntico, pero su uso no lo es. El problema computacional puede formularse entonces en dos etapas: primero, representar relaciones de semejanza entre textos y palabras; luego, hacer que esa representación sea sensible al contexto concreto en el que cada palabra aparece.

1. SIMILITUD Y CONTEXTO

Dos problemas complementarios del significado

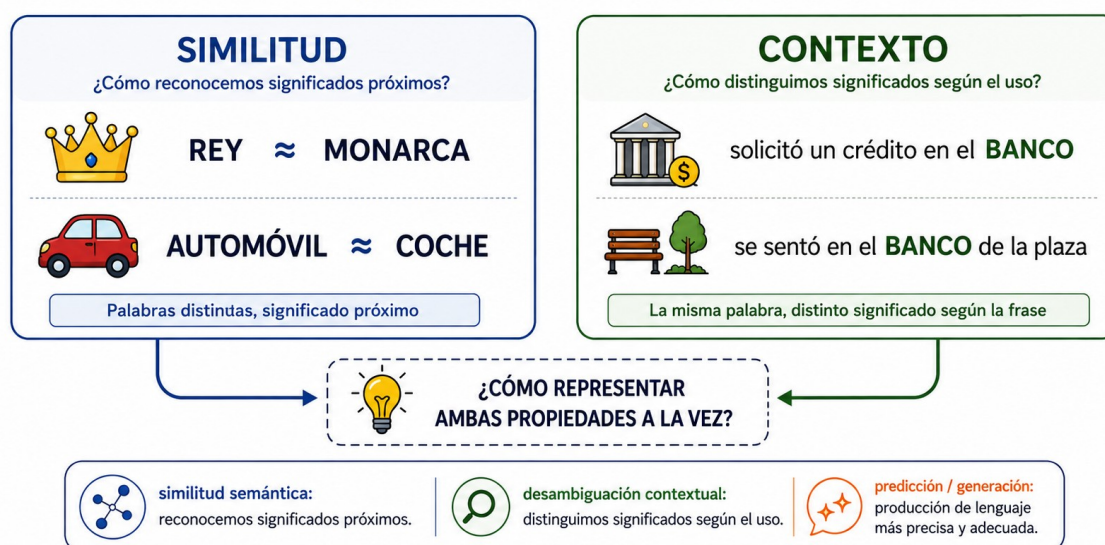


Figura 1. La figura presenta la similitud y el contexto como dos problemas complementarios del significado: reconocer proximidades semánticas entre palabras distintas y distinguir significados diferentes de una misma palabra según el uso. Fuente: elaboración propia a partir del marco conceptual de Wittgenstein (1988) y de la semántica distribucional.

La evolución que se presenta a continuación no debe leerse como una secuencia en la que una técnica elimina por completo a la anterior. TF-IDF continúa siendo útil para recuperación de información; Word2Vec sigue siendo una referencia fundamental para comprender embeddings; BERT y GPT responden a objetivos diferentes. El hilo que los reúne es conceptual: observar cómo se amplía la capacidad de representar semejanzas, incorporar contexto y utilizarlo para desambiguar o predecir lenguaje.

2. De la similitud a la representación

2.1. TF-IDF: representar para comparar

Una primera respuesta computacional al problema de la similitud consiste en transformar los textos en vectores. El modelo de espacio vectorial desarrollado en recuperación de información permitió representar documentos y consultas como puntos dentro de un espacio definido por términos (Salton, Wong y Yang, 1975). En esa lógica, comparar textos deja de depender de una inspección puramente literal y pasa a convertirse en una operación matemática sobre sus representaciones.

TF-IDF combina dos intuiciones simples. La frecuencia de término (TF) aumenta el peso de una palabra cuando aparece con mayor presencia dentro de un documento. La frecuencia inversa de documento (IDF) reduce el peso de los términos que aparecen de manera muy generalizada en toda la colección y aumenta el de aquellos que ayudan a diferenciar unos documentos de otros (Spärck Jones, 1972). En forma resumida:

$$TF-IDF(t,d) = TF(t,d) \times IDF(t)$$

El resultado es un vector de pesos. A partir de allí puede calcularse, por ejemplo, la similitud coseno entre dos documentos. Para una exposición introductoria, la importancia de TF-IDF no reside en memorizar la fórmula, sino en comprender el cambio que introduce: el lenguaje comienza a ser representado en un espacio donde la semejanza puede medirse. El método no produce por sí mismo una comprensión contextual de las palabras. Dos documentos pueden expresar ideas similares con vocabularios diferentes y, a la inversa, compartir muchos términos sin sostener el mismo significado. Sin embargo, establece un principio que será decisivo para el recorrido posterior: representar permite comparar.

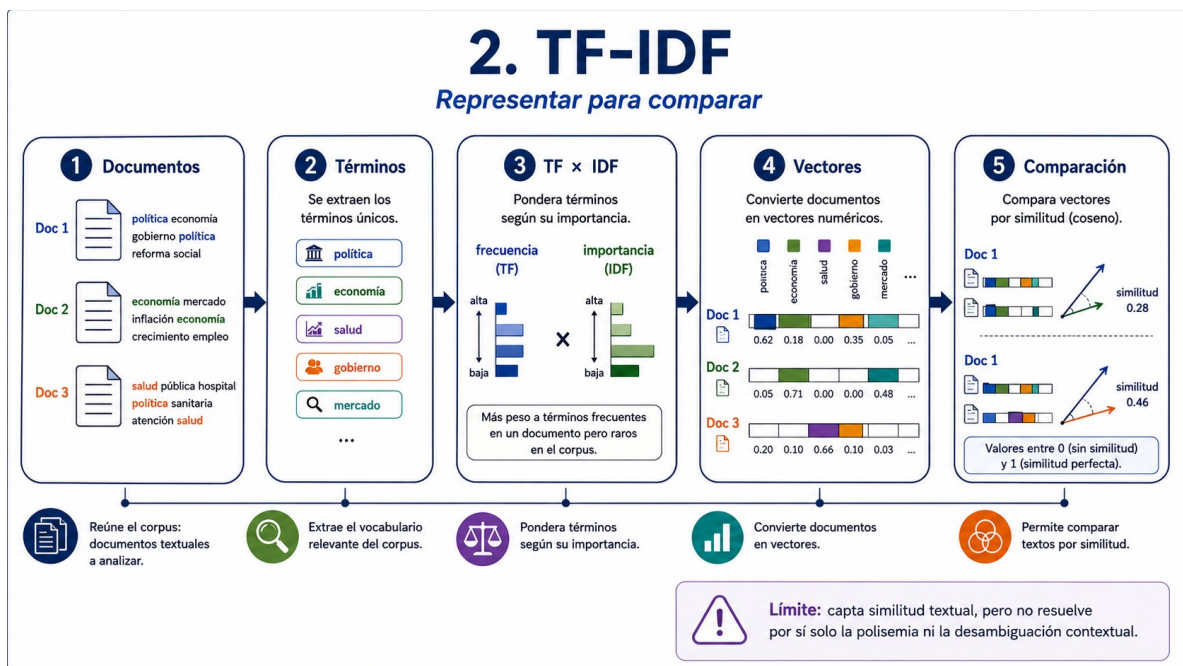


Figura 2. TF-IDF representa documentos como vectores de términos ponderados y permite compararlos mediante medidas de similitud, aunque no resuelve por sí mismo la polisemia ni la desambiguación contextual. Fuente: elaboración propia sobre la base de Spärck Jones (1972) y Salton et al. (1975).

2.2. Word2Vec: aprender relaciones entre palabras

El paso siguiente modifica no solo el tipo de vector, sino también la manera de obtenerlo. Con Word2Vec, las representaciones de las palabras dejan de estar definidas únicamente por reglas explícitas de ponderación y pasan a aprenderse a partir de grandes corpus. Mikolov, Chen, Corrado y Dean (2013) propusieron las arquitecturas CBOW y Skip-gram para aprender vectores densos de palabras a partir de los contextos en los que aparecen. El modelo ya no recibe una definición de rey, mujer, ministro o provincia: aprende regularidades observando qué palabras suelen acompañarse.

Aquí la similitud adquiere una dimensión más cercana a la intuición semántica. Términos empleados en entornos semejantes tienden a ubicarse en regiones próximas del espacio vectorial y ciertas relaciones pueden expresarse como desplazamientos geométricos. El ejemplo clásico —rey menos hombre más mujer aproximadamente igual a reina— resulta pedagógicamente potente porque muestra que el aprendizaje produjo una estructura de relaciones y no solamente una lista de frecuencias. La ecuación no debe leerse como una definición lógica de esos conceptos, sino como una regularidad emergente del espacio aprendido.

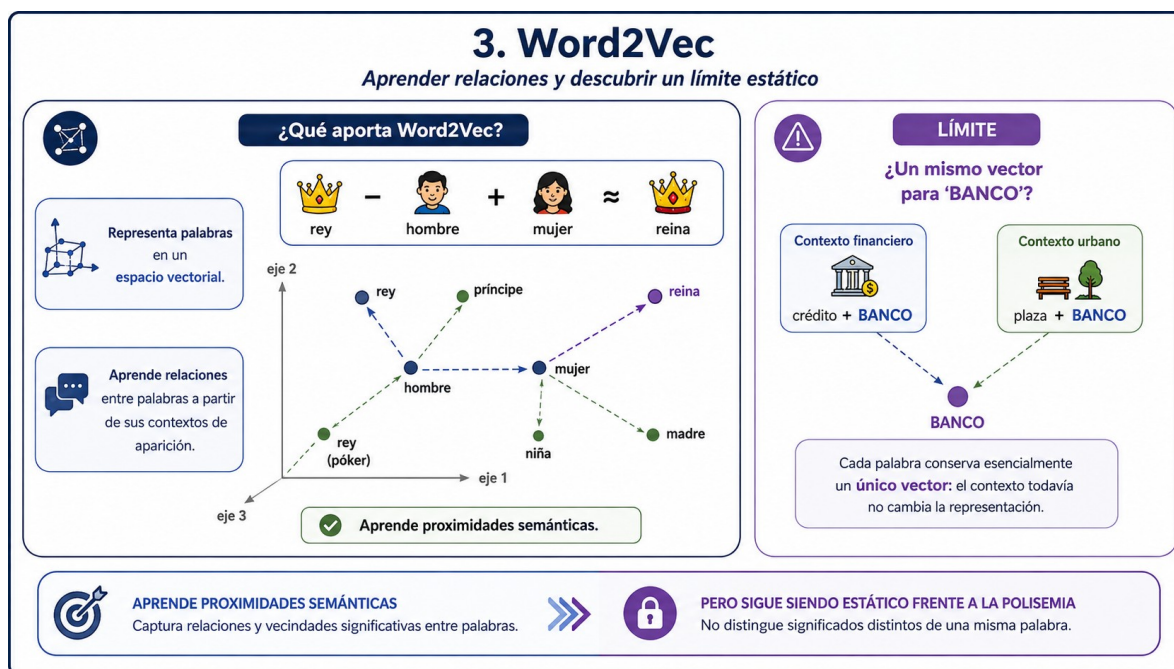


Figura 3. Word2Vec aprende relaciones y proximidades semánticas entre palabras en un espacio vectorial, pero conserva un límite estático frente a la polisemia de términos como “banco”. Fuente: elaboración propia sobre la base de Mikolov et al. (2013).

2.3. El límite del vector estático: cuando la similitud no alcanza

Word2Vec funciona como una bisagra entre la representación estadística y el aprendizaje automático: el sistema aprende vectores a partir de datos. Pero el mismo avance permite ver con mayor claridad un límite. En su forma convencional, una palabra posee un embedding relativamente estable. Banco puede haber sido aprendido a partir de miles de apariciones asociadas tanto con finanzas como con espacios públicos, pero la representación almacenada no se reconstruye desde cero para cada ocurrencia particular.

Ese límite devuelve la discusión al problema inicial. Para reconocer semejanzas generales entre palabras resulta útil contar con una posición aprendida en un espacio semántico; para resolver la polisemia hace falta algo más: que la representación dependa de la frase específica. En otras palabras, ya no basta con preguntar qué palabras suelen parecerse. Debemos preguntar qué relaciones son relevantes en esta secuencia concreta.

3. Del contexto a la desambiguación y la generación

3.1. Transformer: hacer que las palabras se relacionen dentro de la secuencia

La arquitectura Transformer, presentada por Vaswani et al. (2017), ofrece un mecanismo especialmente adecuado para ese cambio de escala conceptual. Su aporte central para esta ponencia puede explicarse sin recorrer toda la ingeniería de la red: cada token puede construir una nueva representación incorporando información de otros tokens de la secuencia mediante self-attention. El contexto deja así de ser una noción externa que el analista agrega después; pasa a intervenir en el cálculo de la propia representación.

Para producir esas relaciones, cada token se proyecta en tres vectores denominados Query, Key y Value. Transformer Explainer propone una analogía especialmente clara para un público no especializado: Query puede pensarse como una consulta en un buscador; Key como aquello que permite decidir qué resultados son relevantes; Value como el contenido que finalmente se recupera. Las comparaciones entre Query y Key producen pesos de atención y esos pesos determinan qué información de los Values se incorpora a la nueva representación. El principio puede resumirse en tres verbos: comparar, ponderar y combinar (Vaswani et al., 2017; Cho et al., 2025).

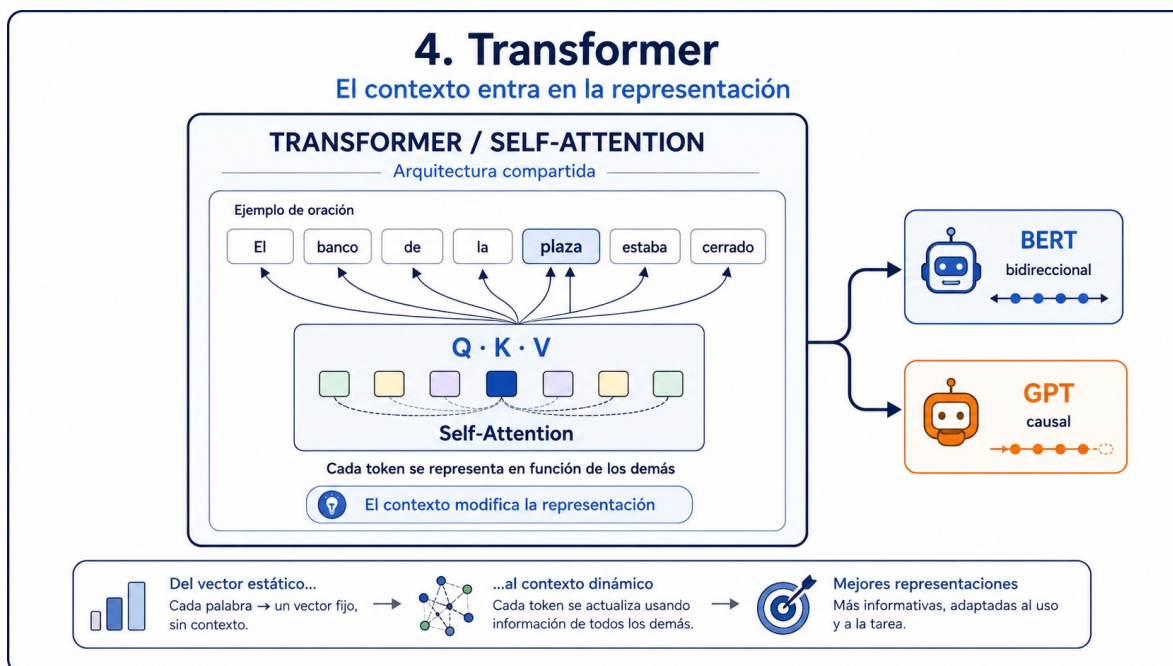


Figura 4. Transformer introduce self-attention como arquitectura compartida: cada token actualiza su representación usando información de los demás, de modo que el contexto entra en la representación. Fuente: elaboración propia sobre la base de Vaswani et al. (2017) y Cho et al. (2025).

La Figura 4 representa el salto que organiza toda la ponencia. En un embedding estático, banco parte de un mismo vector. En un modelo contextual, las palabras que rodean cada ocurrencia modifican la información disponible para construir su representación. El contexto financiero y el contexto espacial dejan de ser únicamente ejemplos que el ser humano distingue externamente: pasan a intervenir en el procesamiento interno del modelo.

3.2. BERT: contexto bidireccional y desambiguación

BERT —Bidirectional Encoder Representations from Transformers— fue diseñado para preentrenar representaciones profundas condicionadas conjuntamente por el contexto situado a izquierda y derecha de cada posición (Devlin et al., 2019). Esta bidireccionalidad resulta especialmente adecuada para recuperar información ausente o construir una representación de una palabra atendiendo a toda la frase. En lugar de asignar simplemente un único significado a banco, BERT produce una representación de banco en ese contexto.

El objetivo de Masked Language Modeling hace visible esta propiedad. Si presentamos “La [MASK] aprobó el proyecto”, el modelo puede utilizar tanto lo que aparece antes de la máscara como lo que aparece después para estimar qué token resulta compatible con esa posición. Desde el punto de vista pedagógico, BERT ofrece entonces una respuesta operativa al problema de la desambiguación: las representaciones no son únicamente parecidas o diferentes en términos generales, sino que se recalculan en función del contexto concreto de uso.

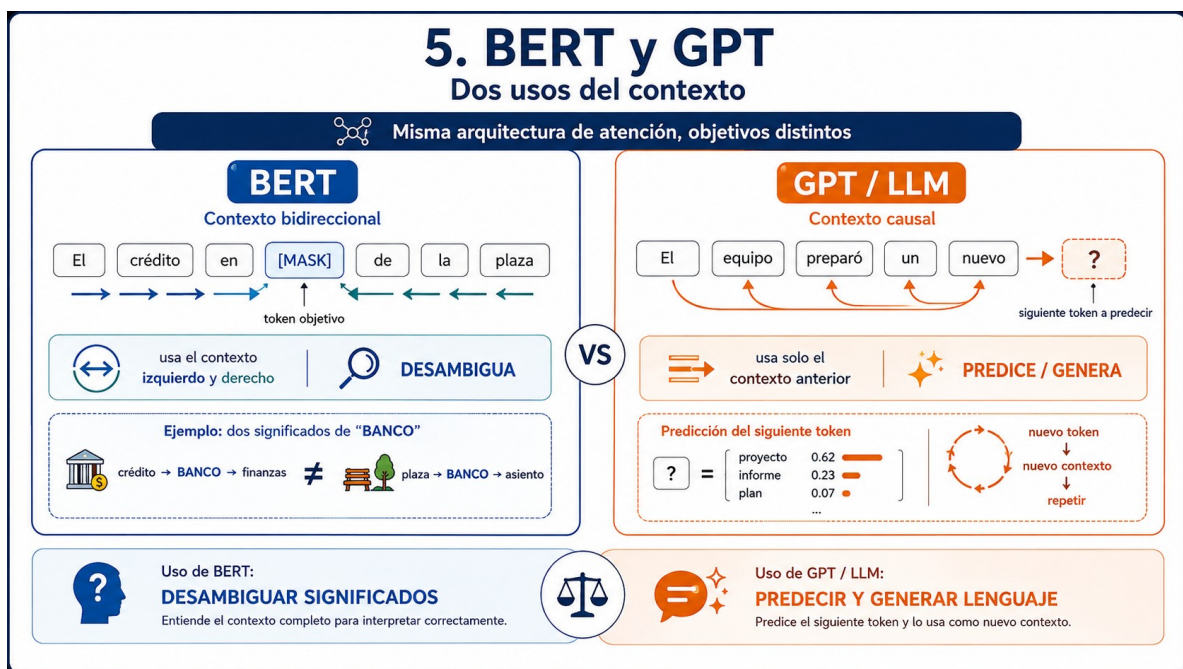


Figura 5. BERT y GPT comparten la arquitectura Transformer, pero emplean el contexto de manera distinta: BERT lo usa bidireccionalmente para desambiguar y GPT de manera causal para predecir y generar lenguaje. Fuente: elaboración propia sobre la base de Devlin et al. (2019) y Radford et al. (2019).

3.3. Del contexto a la predicción: GPT

Una vez explicado que el contexto modifica la representación, podemos preguntar:

¿qué otra cosa podemos hacer con ese contexto además de desambiguar? Predecir.

Esta pregunta introduce la segunda gran rama del paradigma Transformer. GPT no utiliza el contexto del mismo modo que BERT. En un modelo generativo causal, cada posición se calcula a partir de los tokens anteriores y se impide el acceso a los tokens futuros. El objetivo fundamental

consiste en estimar una distribución de probabilidades para el siguiente token. GPT-2 fue entrenado precisamente con ese objetivo: dado un fragmento de texto, predecir qué viene después (Radford et al., 2019).

La diferencia puede expresarse de manera sencilla. BERT pregunta qué palabra encaja en una posición cuando dispone de contexto a ambos lados; GPT pregunta cuál debería ser el próximo token dado lo que ya fue escrito. En ambos casos la predicción depende de representaciones contextualizadas, pero la dirección del acceso al contexto y el objetivo de entrenamiento son diferentes. Por ello no conviene presentar BERT como un paso histórico que “se transforma” en GPT. Son dos desarrollos del paradigma Transformer que permiten observar funciones distintas del contexto: desambiguar y continuar.

3.4. Transformer Explainer: hacer visible la causalidad

Transformer Explainer ocupa un lugar metodológico central en la presentación porque permite convertir esa diferencia conceptual en una experiencia visual. La herramienta ejecuta un GPT-2 Small de 124 millones de parámetros directamente en el navegador y recorre de manera interactiva la tokenización, los embeddings, las proyecciones Q-K-V, la atención causal y la transformación de los resultados finales en probabilidades sobre el vocabulario (Cho et al., 2025). El objetivo no es enseñar la arquitectura completa, sino seguir el camino de una predicción.

La máscara causal resulta especialmente útil para la explicación: cada token puede atender a posiciones anteriores, pero no a posiciones futuras. Después de atravesar los bloques Transformer, la representación final se proyecta sobre el vocabulario; los logits se convierten mediante softmax en una distribución de probabilidades y se selecciona un nuevo token. Ese token se agrega al contexto y el proceso se repite. En ese bucle se encuentra la transición decisiva entre predecir y generar.

3.5. De la predicción causal a los grandes modelos de lenguaje

El principio generativo puede describirse sin introducir los numerosos refinamientos que caracterizan a los sistemas actuales. Una arquitectura Transformer entrenada para predecir el siguiente token puede generar texto repitiendo esa operación de manera autoregresiva. El escalamiento posterior en cantidad de parámetros, datos y capacidad de cómputo amplió notablemente las capacidades observables de estos modelos. GPT-2 mostró que un modelo de lenguaje entrenado sobre grandes corpus podía desplegar comportamientos transferibles a múltiples tareas, y GPT-3 profundizó esa tendencia al escalar el paradigma a 175 mil millones de parámetros y mostrar capacidades few-shot mediante instrucciones y ejemplos expresados en lenguaje natural (Brown et al., 2020).

Para esta ponencia, sin embargo, el punto no es reconstruir toda la historia de los LLM contemporáneos. El interés está en el puente conceptual: los grandes modelos generativos no aparecen porque la computadora haya adquirido una definición fija de cada palabra. Se apoyan en representaciones aprendidas, en mecanismos capaces de relacionar tokens dentro de un contexto y en un objetivo probabilístico de predicción. La capacidad de desambiguar y la capacidad de generar no son idénticas, pero comparten una transformación fundamental: el significado operativo de una representación depende de las relaciones que se construyen en la secuencia.

4. De la teoría a la experiencia

4.1. Primera demostración: observar GPT-2 con Transformer Explainer

La primera experiencia durante la exposición será guiada por el expositor. Transformer Explainer permite que los participantes observen, sin necesidad de programar, cómo un texto se divide en tokens, cómo esos tokens se transforman en embeddings, cómo opera la atención y cómo el resultado desemboca en probabilidades para el siguiente token. Esta demostración cumple una función de puente: antes de pedir al público que experimente con BETO, ofrece una imagen concreta de qué significa afirmar que una predicción se construye a partir del sentido que adquiere la secuencia.

En una ponencia de aproximadamente veinte minutos no se desarrollarán en detalle capas feed-forward, normalización, conexiones residuales ni otros componentes relevantes para una descripción exhaustiva de Transformer. El recorte es deliberado: interesa que el público pueda identificar la secuencia representación → atención → contexto → probabilidad → predicción.

4.2. Segunda experiencia: predecir con BETO y uno o dos [MASK]

La segunda experiencia traslada el principio al español mediante BETO, un modelo BERT preentrenado sobre corpus en español (Cañete et al., 2020). Una interfaz Gradio permite escribir una frase con uno o dos tokens [MASK] y observar los candidatos propuestos por el modelo junto con sus probabilidades. Con una sola máscara se muestra directamente la lógica del Masked Language Modeling. Con dos, la notebook aplica una estrategia secuencial: completa primero una posición, incorpora el término seleccionado al texto y vuelve a calcular la segunda.

La prueba realizada durante el desarrollo de la ponencia mostró con claridad ese efecto. Una primera máscara presentó varios candidatos con probabilidades relativamente distribuidas; después de incorporar la predicción principal al contexto, la segunda posición concentró una probabilidad muy elevada en una alternativa determinada. El interés didáctico no está en que el modelo “acierte” una frase, sino en que el público observe que modificar el contexto modifica la distribución de predicciones.

4.3. Tercera experiencia: extraer entidades con BETO-NER

La tercera experiencia cambia deliberadamente la pregunta. Ya no se consulta qué palabra falta, sino qué información puede extraerse del texto. Para ello se utiliza una versión de BETO ajustada para Named Entity Recognition, capaz de etiquetar personas, organizaciones, lugares y entidades misceláneas. El contraste permite mostrar que una representación contextual preentrenada puede especializarse para objetivos diferentes mediante fine-tuning: predecir una posición y clasificar fragmentos del texto son tareas distintas, aunque comparten una base representacional.

Las pruebas también resultaron útiles para mostrar límites. En una frase de validación, el sistema identificó correctamente México como lugar, pero segmentó el nombre Exequiel en la tabla de salida y clasificó Tecate como organización, aunque el contexto refería a una cerveza. Estos resultados son pedagógicamente valiosos: contextualización no equivale a comprensión humana infalible. Las salidas continúan siendo estimaciones probabilísticas y pueden requerir corrección, evaluación y supervisión.

4.4. Una experiencia introductoria para la gestión pública

Las tres instancias fueron diseñadas para un congreso de ciencias sociales y administración pública, no para un curso de programación. La interfaz Gradio permite que los asistentes ingresen desde sus teléfonos mediante un código QR y experimenten con frases propias. La intención es que conceptos normalmente presentados como arquitectura de redes neuronales se vuelvan observables mediante operaciones sencillas: comparar, atender al contexto, predecir y extraer.

En el ámbito público, estas capacidades pueden formar parte de sistemas de clasificación documental, recuperación de información, identificación de personas, organismos o lugares, procesamiento de consultas y extracción de datos desde corpus administrativos. La demostración no pretende presentar BETO como una solución terminada para esas tareas. Su función es mostrar el principio sobre el que pueden construirse aplicaciones especializadas y, al mismo tiempo, hacer visibles sus límites. En entornos institucionales, la utilidad depende del problema, de los datos, del ajuste del modelo y de procedimientos de validación proporcionales a las consecuencias de una clasificación incorrecta.

Conclusiones

El recorrido propuesto puede resumirse volviendo a la pregunta inicial: ¿cómo representar que dos expresiones se parecen y, a la vez, que una misma palabra cambia de significado según el contexto? TF-IDF ofrece una primera operación vectorial para comparar documentos a partir de términos ponderados. Word2Vec desplaza el problema hacia representaciones aprendidas en las que las relaciones entre palabras adquieren forma geométrica. Ambos pasos hacen visible la potencia de la similitud, pero también su límite cuando aparece la polisemia.

Transformer introduce el mecanismo que permite que las relaciones entre tokens modifiquen las representaciones dentro de una secuencia. BERT explota esa posibilidad de manera bidireccional y ofrece una forma especialmente clara de comprender la desambiguación contextual. El ejemplo de banco sintetiza el cambio: ya no interesa solamente aprender un vector general para la palabra, sino construir una representación diferente en “crédito en el banco” y en “banco de la plaza”.

**Si el contexto puede modificar una representación para desambiguar,
también puede utilizarse para predecir qué lenguaje resulta compatible con ese
contexto.**

GPT hace visible esa segunda posibilidad mediante atención causal y predicción del siguiente token. La generación surge cuando esa predicción se repite: el token producido se convierte en parte del nuevo contexto y condiciona la predicción posterior. El escalamiento de ese paradigma constituye uno de los fundamentos de los grandes modelos de lenguaje generativos. La evolución central de la ponencia no es, por tanto, una simple sucesión de nombres de algoritmos, sino una ampliación de aquello que puede representarse y hacerse con el contexto.

Las experiencias con Transformer Explainer y BETO cierran el recorrido desde la teoría hacia la observación. La primera permite ver cómo GPT-2 transforma una secuencia en probabilidades; la segunda permite comprobar cómo una frase condiciona las predicciones de [MASK]; la tercera muestra cómo las representaciones contextuales pueden reutilizarse para extraer entidades. Los

errores observados forman parte del mismo aprendizaje, porque recuerdan que probabilidad no equivale a certeza y que la aplicación en gestión pública exige evaluación y supervisión.

En ese sentido, la intuición inicial recuperada de Wittgenstein conserva su valor como marco del problema: el significado no se agota en la palabra aislada. Los desarrollos computacionales presentados no confirman una teoría filosófica ni derivan de ella linealmente; muestran, desde otro plano, hasta qué punto representar relaciones de uso y contexto se volvió central para que una máquina pueda comparar, desambiguar, predecir y generar lenguaje.

BIBLIOGRAFÍA

- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., ... Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
- Cañete, J., Chaperon, G., Fuentes, R., Ho, J.-H., Kang, H., & Pérez, J. (2020). Spanish pre-trained BERT model and evaluation data. PML4DC at ICLR 2020.
- Cho, A., Kim, G. C., Karpekov, A., Helbling, A., Wang, Z. J., Lee, S., Hoover, B., & Chau, D. H. (2025). TRANSFORMER EXPLAINER: Interactive learning of text-generative models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28), 29625–29627. <https://doi.org/10.1609/aaai.v39i28.35347>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Firth, J. R. (1957). A synopsis of linguistic theory, 1930–1955. In *Studies in Linguistic Analysis* (pp. 1–32). Philological Society.
- Harris, Z. S. (1954). Distributional structure. *Word*, 10(2–3), 146–162. <https://doi.org/10.1080/00437956.1954.11659520>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv*. <https://arxiv.org/abs/1301.3781>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 3111–3119.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI*.
- Romero, M. (s. f.). bert-spanish-cased-finetuned-ner [Modelo de aprendizaje automático]. Hugging Face. <https://huggingface.co/mrm8488/bert-spanish-cased-finetuned-ner>
- Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613–620. <https://doi.org/10.1145/361219.361220>
- Spärck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1), 11–21. <https://doi.org/10.1108/eb026526>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- Wittgenstein, L. (1988). *Investigaciones filosóficas* (A. García Suárez & U. Moulines, Trans.). Crítica. (Trabajo original publicado en 1953).