

Modelos vectoriales de representación semántica

En este tema veremos:

- los fundamentos de la semántica distribucional como modelo teórico,
- la representación vectorial de las relaciones semánticas distribucionales y los factores que lo determinan,
- los conceptos de distancia y similitud como métodos de interpretación, y las medidas básicas.

6.1. Lecturas

Para completar este tema, se recomiendan las siguientes lecturas (opcionales):

- El capítulo 6 “Vector Semantics and Embeddings” de Jurafsky and Martin (2023) *Speech and language processing* completa todos los aspectos aquí planteados. Sobre todo el punto 6.8, donde se explica cómo se calculan los *word embeddings*. Disponible aquí: <https://web.stanford.edu/~jurafsky/slp3/6.pdf>
- Peter D. Turney y Patrick Pantel (2010) “From Frequency to Meaning: Vector Space Models of Semantics” en *Journal of Artificial Intelligence Research*, 37, págs. 141-188. DOI: <https://doi.org/10.1613/jair.2934>. Disponible aquí <https://www.jair.org/index.php/jair/article/view/10640>.

Este artículo explica bien cómo un vector de frecuencias puede representar el significado de una palabra. El trabajo es ya antiguo, de 2010, anterior al desarrollo de los modelos neuronales y los *word embeddings*, por lo que hay aspectos hoy día ya desfasado. Son de interés, sobre todo, los primeros epígrafes.

6.2. Introducción

Los modelos semánticos vectoriales son una aproximación formal para la representación de la semántica de un texto totalmente diferente a los modelos simbólicos vistos anteriormente. La diferencia viene dada tanto por el modelo teórico subyacente como por el formalismo en sí mismo:

- El modelo teórico subyacente es la semántica distribucional, que asume que el significado de una palabra deriva de su uso concreto en un contexto determinado.
- El formalismo de representación son espacios vectoriales multidimensionales, que es una representación propia del modelo conexionista.

6.3. La semántica distribucional

El modelo semántico distribucional tiene su origen en la teoría distribucional del lenguaje, que se inició a mediados del s. XX y que se caracteriza, sobre todo, por la importancia que le da al contexto en todos los niveles lingüísticos.

En el capítulo anterior se vio que el significado se podía caracterizar de diferentes maneras. Así, el modelo de base lógico considera el significado de una palabra como una unidad atómica (el significado de “casa” sería CASA, o como variable $casa(x)$); el modelo lexicográfico considera que el significado de una palabra es también una unidad discreta y se representa mediante la definición de un diccionario; o el modelo léxico de WordNet, que considera que el significado de las palabras es un nodo finito (el *synset*) dentro de una red de relaciones léxicas tipo hiperonimia, hponimia, etc.

El planteamiento semántico-vectorial es diferente. No se considera el significado como una unidad atómica y finita. Más bien se considera el significado a partir de las relaciones contextuales que una palabra tiene en los contextos donde aparece. Este modelo está basado en los siguientes planteamientos lingüísticos [9]:

1. La idea de Wittgenstein de “meaning just is use” [32], es decir, que más allá de diccionarios, estudios y academias; una palabra tendrá el significado que le den los hablantes cuando la usen en textos reales. El significado es simplemente el uso que se les dé a las palabras.
2. El concepto de *collocation* de Firth (1957) y su idea de que

“you shall know a word by the company it keeps”. [13]

Dicho con otras palabras, según esta idea es posible conocer cómo son las palabras, incluido su significado, estudiando con qué otras palabras puede aparecer, es decir, estudiando su contexto. En español podrías decir de las palabras aquello de “dime con quién andas, y te diré quién eres”.

3. La hipótesis distribucional de [16], que dice

“words will occur in similar contexts if and only if they have similar meanings”.

O dicho al contrario, un significado similar implica un contexto similar. De tal manera que podríamos determinar la similitud semántica entre dos palabras a partir de la similitud entre sus contextos.

En realidad, esto de determinar el significado de una palabra a partir de las palabras del contexto es algo que hacemos constantemente. Lee las siguientes oraciones, ¿qué significado tiene XXX en cada una?

- Mañana iré al XXX a firmar la hipoteca, y ya de paso sacaré dinero del cajero.
- He intentado ponerme los XXX de mi hermano pero me vienen pequeños: mis pies son muy grandes y necesito una talla más.

Esta idea de determinar palabras a partir de huecos será clave para la creación de los grandes modelos de lenguaje como BERT o GPT, como se verá en próximos temas.

De todo ello se deduce que una representación formal del contexto de una palabra es una representación computacional de su significado, pues ese contexto está en función de su uso real, especifica las palabras con las que suele aparecer la palabra objeto y permite determinar otras palabras similares a partir de la similitud entre contextos.

Esto es precisamente lo que hacen los modelos semánticos vectoriales. En este caso, para representar de manera formal el contexto, se utilizan vectores.

6.4. Espacio vectorial como modelo de representación formal

La semántica vectorial, por tanto, es una representación formal del significado de las palabras mediante vectores.

Este formalismo se basa, por tanto, en los modelos de espacio vectorial y permiten realizar operaciones sobre vectores y matrices propias del álgebra lineal. También está en la base de las aproximaciones cuánticas al lenguaje (*quantum semantics* en este caso) [10, 21] y el análisis semántico en términos geométricos (cálculo de similitud semántica basadas en *distancias*) [31].

Origen

El origen de la representación vectorial del significado en PLN está en la tarea de recuperación de información (*Information Retrieval*: IR) como son, por ejemplo, los buscadores de internet.

Como ya sabrás, estos sistemas, dada una consulta (formada por una o más palabras o términos), obtienen una lista de documentos ordenados de mayor a menor relevancia según la consulta.

Para ello, antes de la búsqueda en sí, la colección de documentos se transforma en una matriz término-documento. Esta matriz, en su forma más básica, considera el texto como una bolsa de palabras (*bag of words*), es decir, no toma en consideración las relaciones entre las palabras (ver temás anteriores): el texto es solo un conjunto no estructurado de palabras. En la matriz, cada documento es una columna y cada línea un término (palabra). Los valores de la matriz (las celdas) son la frecuencia de cada término en cada documento.

Así, por ejemplo, si asumimos los dos siguientes documentos:

- $\text{doc1} = \{\text{casa}, \text{madera}, \text{mesa}\}$
- $\text{doc2} = \{\text{papel}, \text{rama}, \text{madera}\}$

Se podría crear una matriz como la representada en el Cuadro 6.1.

Con matrices como ésta, para un sistema de RI es posible seleccionar la columna más relevante (texto) a partir de los valores de las filas (palabras) de la consulta. Pero esto ya no interesa aquí. Lo relevante para PLN es cómo esta matriz puede representar el significado de las palabras.

Representación vectorial del significado

Cada línea de este tipo de matrices es un vector que captura, dada una palabra, la relación contextual de ésta con el resto de palabras del corpus. Los

	Doc1	Doc2	...	Doc n
casa	1	0
madera	1	1
mesa	1	0
papel	0	1
rama	0	1

Cuadro 6.1: Matriz término-documento

	doc1	doc2
car	7	6
taxi	5	6
train	6	1

Cuadro 6.2: Matriz término documento (2)

valores del vector muestran la relación contextual (peso, relación, implicación, ...) de esa palabra con el resto: en qué medida dos palabras comparten contexto y con qué frecuencia (peso). En términos distribucionales, ese vector está capturando de manera formal el significado distribucional la palabra.

En definitiva, un vector de este tipo captura la semántica contextual/distribucional de la palabra (*token* o lema) al representar el número de veces (frecuencia) que la palabra aparece en cada contexto (en este caso, el contexto es el documento entero).

Así, de la matriz término - documento del Cuadro 6.2, el significado de cada palabra sería el vector contextual:

$$car = (7, 6)$$

$$taxi = (5, 6)$$

$$train = (6, 1)$$

por lo que realmente ahora el corpus es:

$$corpus = \begin{pmatrix} 7 & 6 \\ 5 & 6 \\ 6 & 1 \end{pmatrix}$$

Esto se puede representar en un espacio euclídeo (plano o lineal) mediante coordenadas cartesianas: los valores del vector se proyectan en los ejes de coordenadas, siendo la abscisa x el documento 1 (primera columna de la matriz) la ordenada y el documento 2 (segunda columna de la matriz). Así,

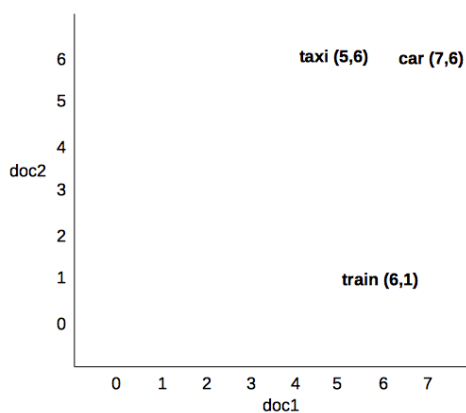


Figura 6.1: Representación vectorial del significado léxico

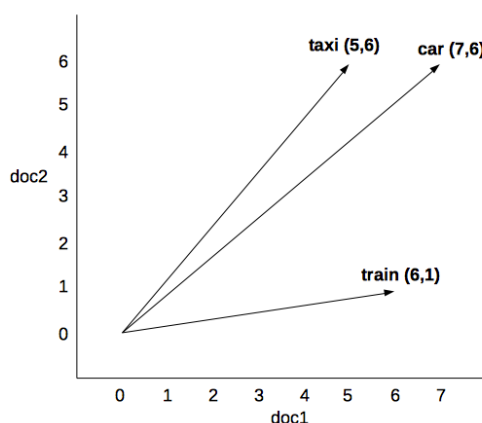


Figura 6.2: Representación vectorial del significado léxico

el significado de cada palabra es un punto en el espacio euclídeo (Figura 6.1). Si unimos ese punto con el origen $(0,0)$, se obtiene el vector en términos matemáticos (Figura 6.1).

En este caso, para poder visualizarlo, el corpus está formado por solo dos textos (documentos) y por tanto el espacio es bidimensional. En aplicaciones reales hay n documentos, por lo que el espacio es multidimensional.

Estos vectores n -dimensionales que representan el significado contextual de las palabras ya se podría denominar *word embeddings*, que son la representación semántica que utilizan los modernos modelos neuronales. Pero para ser operativos es necesario reducir la dimensionalidad de la matriz y su dispersión (cantidad de ceros). Estos dos problemas serán tratados en próximos

temas.

La representación del significado varía según se diseñe el espacio vectorial. Esta representación depende de tres factores:

1. Representación del contexto (dimensiones)
2. Representación de las palabras (filas)
3. Valores o pesos de cada palabra en cada contexto.

Representación del contexto (dimensiones)

Cada contexto de la palabra será una dimensión de la matriz. El problema es cómo delimitar este contexto: ¿cuántas palabras forman el contexto?, ¿dónde está el límite del contexto?

En el modelo de matriz término-documento que se utiliza en recuperación de información el contexto es todo el documento (todo el texto) porque son documentos lo que se quiere recuperar, pero se puede limitar a recuperación de pasajes, párrafos, etc.; es decir, se puede limitar el contexto según se quiera una representación u otra. Otras opciones de representación del contexto con motivación lingüística podrían ser:

- la oración,
- una ventana deslizante (un conjunto de palabras delante y detrás de la término),
- el párrafo o cualquier otra unidad textual,
- el capítulo,
- etc.

Por otro lado, además de la matriz término-documento que hemos visto (donde las columnas representan documentos y las filas palabras), se puede crear otro tipo de matriz: la llamada matriz de co-ocurrencias o matriz término-término. En estas matrices (normalmente cuadradas), tanto las columnas como las filas representan palabra, y los valores la relación entre esas dos palabras. Por ejemplo, esos valores pueden representar en cuántos contextos aparecen juntas esas dos palabras, como en el Cuadro 6.3:

Según esta matriz, la palabra “car” aparece el mismo contexto de la palabra “red” en cinco ocasiones (de un total de N contextos), no coincide en ningún contexto con la palabra “readable” y solo en uno con la palabra “blue”. “book”, por su parte, aparece tres veces en el mismo contexto de “red”, seis

	red	readable	blue
car	5	0	1
book	3	6	0

Cuadro 6.3: Matriz de coocurrencia

en el mismo contexto de “readable” y ninguna con “blue”. En ocasiones estas matrices son cuadradas porque tienen los mismo términos en las filas y en las columnas.

Se pueden plantear otros tipos de matrices. Turner y Pantel [30], por ejemplo, plantean una matriz *Pair-Pattern* donde las filas son parejas de palabras $X : Y$ (“carpenter:wood”) y las columnas son relaciones entre palabras co-ocurrentes (“X cut Y”).

Sea como sea el tipo de matriz, es muy relevante dónde se sitúa el límite del contexto (el documento, el párrafo, la oración, etc.) pues de este límite se podrán obtener representaciones muy precisas en matrices muy dispersas (es decir, con muchas celdas con valor 0), como ocurre con contexto muy pequeños, o representaciones menos precisas pero con matrices (algo más) densas (con menos cantidad de celdas con valor 0). Luego se comentará más sobre el problemas de las matrices dispersas.

Representación de las palabras

Hasta ahora hemos estado hablando de “palabra” o “término”, pero como ya se vio en temas anteriores el concepto de “palabra” es muy vago. Una matriz será más o menos representativa según se defina la palabra. Algunas opciones son (según vimos) pueden ser:

- el *token*,
- la raíz o *stem*,
- el lema,
- el lema más la categoría gramatical,
- los *tokens* pero eliminando *stopwords*,
- solo *tokens* de determinadas categorías gramaticales (solo nombres, o solo verbos, o solo adjetivos, etc.)
- el lema más su dependencia sintáctica,
- u otros casos.

Algunos de estos casos, como imagino ya sabrás, requieren procesar el corpus previamente con técnicas específicas de PLN, como lo que se vio en temas anteriores. Esto es complejo si la colección es muy amplia. En estos casos se utiliza el *token* o el carácter.

Cálculo de los valores o pesos

Finalmente, el modelo semántico vectorial puede ser más o menos representativo según se mida la relevancia (o peso) de la palabra en cada contexto.

El caso más simple para medir la relevancia de una palabra en un contexto es calcular la frecuencia ponderada: número de veces que la palabra aparece en el contexto, normalizado por el tamaño del contexto. Este modelo tiene, sin embargo, diversos problemas:

- Es muy dependiente del tamaño del contexto, que como hemos visto antes no está claro cómo limitarlo. En contexto pequeños se trabajaría con valores muy bajos (ceros y unos prácticamente).
- No discrimina la importancia real de cada palabra en el contexto, dado que hay palabras que siempre tienen frecuencias muy alta (como palabras de categorías gramaticales cerradas -artículos, preposiciones, conjunciones, etc.-, o nombres de uso muy común) frente a otras que siempre tienen frecuencias bajas.
- Sobre estas últimas, el caso extremo es el fenómeno del *hapax legomenon*:¹ las palabras que solo aparecen una vez en todo el corpus. El problema es que suelen ser la mayoría de las palabras: solo aparecen una vez, o con una frecuencia muy baja.

Una solución elegante para determinar la relevancia de una palabra por su frecuencia sin caer en estos problemas es el famoso valor TF/IDF que pasamos a explicar a continuación.

TF/IDF: term frequency / inverse document frequency (Sparck Jones, 1972)

La idea intuitiva que subyace a este valor es que las palabras de uso muy común (aquellas que aparecen con alta frecuencia en prácticamente todos los documentos) no son discriminativas para determinar la importancia del documento. Tienen por tanto poca relevancia en su documento y por tanto su

¹Ver https://en.wikipedia.org/wiki/Hapax_legomenon

valor debe ser bajo. Las palabras que realmente son relevantes en un documento, las que lo caracterizan, son aquellas que tiene una frecuencia relativamente alta en un documento pero, al mismo tiempo, tiene una frecuencia relativamente baja o nula en el resto de documentos. Esto es lo que intenta modelar TF/IDF: dar más peso a las palabras con frecuencia relevante en unos documentos pero no en la totalidad de la colección de textos.

TF/IDF son las siglas de “frecuencia del término por la frecuencia inversa del documento”. Así, en la fórmula nos encontramos con:

- *Term frequency* ($tf(w, d)$): frecuencia relativa de una palabra w en un documento d
- *Document frequency* ($df(t)$): cantidad de documentos donde aparece una determinada palabra w .
- *Inverse document frequency* ($idf(d, D)$): el valor determinante para saber la relevancia del documento es la inversa de la frecuencia de documentos donde aparece. Por tanto, se divide la cantidad toda de documentos N en la colección D entre la frecuencia del documento $df(t)$. Hay varias formas de obtener este valor. La más sencilla es logarítmica, tal que $idf(d, D) = \log \frac{N}{df}$

Así, el valor tf-idf de la palabra w en un documento d en una colección de documentos D es:

$$tfidf(w, d, D) = tf(t, d) \cdot idf(t, D)$$

PPMI: Point Wise Mutual Information (Church y Hanks 1990)

Otra alternativa clásica para medir el peso contextual de una palabra que ha sido muy utilizada en PLN es el *Point Wise Mutual Information* o PPMI. Así como tf-idf es la medida estándar para medir la relevancia de las palabras en matrices término-documento, PMMI es la medida que se suele utilizar en matrices de coocurrencia término-término.

La intuición detrás de PPMI es que la coocurrencia de dos palabras en el mismo contexto es relevante en la medida que podamos saber la posibilidad de que ambas palabras coocurran por casualidad. Si no coocurren en el mismo contexto por casualidad, es que esa coocurrencia es motivada y por tanto relevante.

Así, PMMI mide la probabilidad de que dos palabras aparezcan en el mismo contexto, y lo divide por la probabilidad de aparición de cada palabra por separado:

$$PPMI(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)}$$

Hoy día tf-idf y PPMI son medidas clásicas para la representación vectorial del significado. Luego se verán otras propuestas para determinar la relevancia de cada palabra en el contexto donde aparece; pero antes hay que tratar el gran problema de los modelos semánticos vectoriales: la matriz dispersa.

Matriz dispersa y matriz densa

Dada las características de los idiomas, este tipo de matrices de coocurrencias (bien sean término-término o término-documento) que miden las relaciones contextuales entre palabras son siempre matrices muy dispersas (*sparse matrix*), es decir, son matrices en las que la mayoría de los valores son cero (o uno). Lo normal en un idioma es que dos palabras no compartan contexto. Las palabras que comparten contexto entre ellas son pocas, por lo que lo normal es que el valor entre dos palabras sea cero. Esto es un problema tanto desde punto de vista matemático como computacional: se generan estructuras muy grandes pero muy poco informativas.

La solución a este problema es transformar la matriz dispersa en una matriz densa (*dense matrix*), es decir, una matriz sin ceros donde todas las relaciones entre palabras tengan valor superior a 0. Este problema ha sido el principal interés en la investigación en los últimos treinta años. Vamos a comentar tres soluciones que han tenido especial relevancia.

Una primera solución fue *Latent semantic analysis*² o LSA (Landauer y Dumais 1997). Esta aproximación consigue reducir una matriz dispersa en una matriz densa de 300 dimensiones mediante su descomposición en valores singulares (*singular value decomposition*).³ Lo interesante de la matriz resultante no es solo que sea una matriz densa; sino que esa matriz densa, además de mantener las relaciones contextuales entre palabras, muestra relaciones semánticas “latentes”: relaciones semánticas entre palabras que a simple vista no se detectan. LSA, así, supuso un avance en semántica vectorial en las tres áreas de conocimiento implicadas: matemática, computación y lingüística.

Años más tarde se propuso *Latent Dirichlet Allocation*⁴ o LDA, mediante el cual se pueden detectar temas o *topic* en amplios corpus de manera no supervisada.

Finalmente, la búsqueda de matrices densas y la optimización de la representación contextual mediante vectores ha llevado a los *skip grams*, que es la

²https://en.wikipedia.org/wiki/Latent_semantic_analysis

³https://en.wikipedia.org/wiki/Singular_value_decomposition

⁴https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation

base de *Word2Vec* y de donde derivan los *word embeddings* y los modelos neuronales actuales. Para determinar la relevancia contextual entre dos palabras, la idea principal de los *skip gramms* es entrenar un clasificador con regresión logística que aprenda si una palabra formar parte o no del contexto de otra palabra. Como solo aprende si dos palabras comparten o no contexto, ese entrenamiento no necesita un corpus anotado a mano: basta con una amplia colección de documentos (cuanto más grande mejor). Y al final lo de menos es el clasificador: lo importante son los pesos que ha aprendido para cada palabra. Ese es su vector contextual o *word embedding*, que ha demostrado tener gran capacidad de representación semántica. Esto es el inicio del *deep learning* y del PLN moderno basado en redes neuronales, que se verá en próximos temas.

6.5. Interpretación semántica: distancia y similitud.

El vector de una palabra en sí mismo no tiene un significado como podría tenerlo, por ejemplo, una definición. Es una representación semántica más de tipo conexionsita que simbólico. Decir que el significado de la palabra “casa” es un vector tal que $\{1, 4, 0, 0, 1\}$ es como no decir nada: ese vector no tiene relación con un concepto cognitivo, sólo indica la relevancia de la palabra en cada contexto (el uso de la palabra).

¿Cómo se realiza, entonces, la interpretación de una palabra y oración en el modelo semántico vectorial? La interpretación en esta aproximación vectorial a la semántica distribucional se realiza por relaciones de *similitud* entre palabras, oraciones, fragmentos o documentos. Dos palabras con vectores contextuales similares implica que ambas palabras tienden a aparecer en los mismos contexto, y por tanto su significado está relacionado. Dos palabras cuyos vectores contextual sean muy diferentes implica que son palabras con significado dispar. Cualquier aplicación de semántica vectorial debe pensarse en términos de similitud entre palabras, grupos de palabras, textos, etc. y no tanto como una interpretación sustancial.

La similitud se calcular según la distancia entre los vectores en el espacio vectorial: a menor distancia entre vectores, mayor similitud semántica. Si bien hay diferentes medidas para calcular la distancia entre vectores, las más utilizada es la distancia del coseno, que mide el ángulo entre dos vectores ambos con origen en 0, 0:

$$\cos(a, b) = \frac{ab}{||a|| ||b||}$$

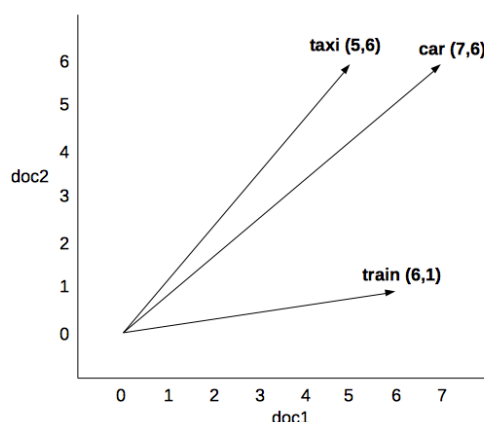


Figura 6.3: Representación vectorial del significado léxico

6.6. Conclusiones

En este capítulo se ha visto la idea general de la semántica vectorial, según la cual el significado de una palabra depende de su uso y de los contextos donde se usa. Se ha visto cómo se representa formalmente ese significado distribucional mediante vectores, cuyos valores miden el peso de la palabra en cada uno de los contextos (bien sean textos, oraciones, etc.). Finalmente se ha comentado brevemente el proceso de interpretación distribucional, que se basa en determinar la similitud entre vectores a partir de su distancia en el espacio multidimensional.

6.7. Situación actual

De aquí derivan los *word embeddings* que, junto con las redes neuronales, forman el PLN moderno. De todo esto se hablará en las próximas sesiones.

6.8. Herramientas y recursos

Para crear espacios vectoriales y calcular similitudes (sin entrar por ahora en redes neuronales):

- GENSIM⁵

⁵<https://radimrehurek.com/gensim/>

- NLTK⁶
- Pattern⁷
- SpaCy⁸

⁶<http://www.nltk.org/>

⁷<http://www.clips.ua.ac.be/pattern>

⁸<https://spacy.io/>

Temas abiertos en PLN

7.1. Introducción

Si bien los modelos neuronales han conseguido logros bastante impresionantes en el procesamiento automático del texto, las lenguas naturales son bastante más complejas de lo que la tecnología actual puede procesar. Hay, así, diferentes cuestiones que aún no tienen una respuesta válida por parte del PLN. En esta sección se exponen algunas de ellas

7.2. Más allá de la alquimia

Una de las grandes críticas a los modelos neuronales es su falta de transparencia. Son cajas negras: resultan muy complejo explicar qué hacen, cómo procesan la información. En tareas concretas (*end-to-end*) funcionan bien, pero no se sabe cómo, ni cuándo fallarán. Esto los convierte en una especie de alquimia computacional, alejada de la claridad y “explicabilidad” necesaria en la vertiente más científica de la inteligencia artificial.¹

Este concepto de “alquimia” lo plantean Church y Liberman en este artículo [8], y proponen líneas de trabajo futuras:

K. Church and M. Liberman (2021) “The Future of Computational Linguistics: On Beyond Alchemy”, *Front. Artif. Intell.*, vol. 4, 2021 <https://www.frontiersin.org/articles/10.3389/frai.2021.625341/full>.

¹Me refiero aquí con “científica” a que, más que soluciones a problemas, buscar explicaciones a fenómenos de la naturaleza (humana en este caso). No tiene ningún tipo de connotación ni positiva ni negativa.

7.3. Composicionalidad

Como se vio en temas anteriores, uno de los grandes principios de la aproximación simbólica a la interpretación semántica de un texto es el principio de composicionalidad. Este principio, sin embargo, solo es aplicable a los modelos simbólicos (de base lógica). No casa bien con la representación geométrica (vectorial, distribucional) del significado en la que se basan los modelos neuronales modernos. Kornai [18] considera ésta como “non-compositional semantics”.

El problema se puede formular del siguiente modo. Un *word embedding* representa el significado de una palabra mediante un vector. Una oración estará formada por tantos vectores como palabras tenga. El principio de composicionalidad dice que el significado de un texto está en función del significado de las palabras y sus relaciones. ¿Cómo se podría representar el significado de la oración con un único vector? ¿Suma ponderada de vectores? ¿Y cómo se representaría con vectores unidades textuales?

Modelos como SentenceBERT² toman esta idea de suma de vectores para representar con un solo vector el significado oracional a partir de *word embeddings*. Una especie de *sentence embedding*. Ver <https://arxiv.org/abs/1908.10084>

En 2023 se ha presentado un modelo (en realidad es un meta-modelo) capaz de hacer generalizaciones vía composicionalidad [19]. Lectura ésta necesaria para profundizar en este tema:

<https://www.nature.com/articles/s41586-023-06668-3>

Otra alternativa es la representación composicional propuesta por los modelos cuánticos. Ver [10]

7.4. Metáforas

La metáfora no es (solo) un fenómeno literario, sino un mecanismo cognitivo para nombrar la realidad [20]. Utilizamos metáforas, entre otras cosas, para tratar con conceptos muy abstractos. Siempre que hablamos del tiempo (como entidad abstracta), por ejemplo, lo hacemos con términos propios del campo del dinero (que es algo concreto), con expresiones como “ahorrar tiempo”, “perder el tiempo”, o la expresión “el tiempo es oro”. Este tipo de metáfora, denominada metáfora cognitiva, es ubicua, está presente en cualquier tipo de texto.

La metáfora, por definición, se utiliza un término en un contexto que no le corresponde. Se produce un conflicto entre el significado del término de manera aislada y el significado que asume en el contexto donde aparece. Es

²<https://www.sbert.net/>

así un fenómeno semántico muy difícil de tratar tanto con los modelos distribucionales como con los modelos simbólicos clásicos.

Sobre los problemas actuales para procesar metáforas, véase [15, 24]

7.5. Neural-Symbolic AI

Como no podía ser de otra manera, hay propuestas híbridas para combinar los modelos simbólicos y los modelos conexistas. Una de ellas es la llamada Neural-Symbolic AI en [4]. Según este trabajo:

“The goals of neural-symbolic computation are to provide a coherent, unifying view for logic and connectionism, to contribute to the modelling and understanding of cognition and, thereby, behaviour, and to produce better computational tools for integrated machine learning and reasoning. To this end, logic and network models are studied together as integrated models of computation. Typically, translation algorithms from a symbolic to a connectionist representation and vice-versa are employed to provide either (i) a neural implementation of a logic, (ii) a logical characterisation of a neural system, or (iii) a hybrid learning system that brings together features from connectionism and symbolic artificial intelligence.”

Ver <https://arxiv.org/abs/1711.03902>

7.6. Texto narrativos

Con el texto narrativo pasa algo similar a lo que se comentó sobre las metáforas: es un tipo de texto muy común (por ejemplo, la noticias periodísticas suelen tener estructura narrativa) y especialmente complejo de procesar. Entender un texto narrativo implica detectar los eventos expresados y las relaciones tanto temporales como causales entre ellos.

Desde el punto de vista de la generación automática del lenguaje, es también un tipo de texto difícil. Los grandes modelos de lenguaje actuales como GPT pueden generar texto narrativo, pero éste presenta varios problemas. No son capaces, por ejemplo, de establecer relaciones causales entre eventos, no hacen un plan narrativo general, no hay suspense ni sorpresa (suelen ser narraciones aburridas por evidentes), etc.

El tratamiento del texto narrativo, tanto en su interpretación como, sobre todo, en su generación, es un tema aún por explorar a fondo. Una presentación de los principales problemas puede verse en este reciente trabajo de A. Pipper: [27]

Bibliografía

- [1] E. Agirre and A. Soroa. Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL)*. Association for Computational Linguistics, 2009.
- [2] John L. Austin. *Cómo hacer cosas con palabras: palabras y acciones*. Paidós, Barcelona, 2016 (1962).
- [3] C. F. Baker, C. J. Fillmore, and J. B. Lowe. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*. Association for Computational Linguistics, 1998.
- [4] Tarek R. Besold, Artur d’Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kuehnberger, Luis C. Lamb, Daniel Lowd, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, Hoifung Poon, and Gerson Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation, 2017.
- [5] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. 2019.
- [6] Noam Chomsky. *Aspects of the theory of syntax*. MIT Press, Cambridge, 1965.
- [7] Noam Chomsky. *The Minimalist Program*. MIT Press, Cambridge, 1995.
- [8] K. Church and M. Liberman. The Future of Computational Linguistics: On Beyond Alchemy. *Frontiers in Artificial Intelligence*, (4), 2021.

- [9] Daoud Clarke. A context-theoretic framework for compositionality in distributional semantics. *Computational Linguistics*, 38(1):41—71, 2012.
- [10] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark†. Mathematical Foundations for a Compositional Distributional Model of Meaning. *Linguistic Analysis*, (36), 2010.
- [11] Jacob Eisestein. *Introduction to Natural Language Processing*. MIT Press, Cambridge, 2019.
- [12] C. Fellbau. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, 1998.
- [13] John R. Firth. *Papers in Linguistics. 1934-1951*. Oxford University Press, 1957.
- [14] J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:2–71, 1988.
- [15] Mengshi Ge, Rui Mao, and Erik Cambria. A survey on computational metaphor processing techniques: from identification, interpretation, generation to application. *Artificial Intelligence Review*, 56:1829—1895, 2023.
- [16] Zellig Harris. *Structural Linguistics*. University of Chicago Press, Chicago, 1951.
- [17] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. 2023.
- [18] András Kornai. *Vector semantics*. Springer, 2023.
- [19] B. M. Lake and M. Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623:115—121, 2023.
- [20] George Lakoff and Mark Johnson. *Metáforas de la vida cotidiana*. Cátedra, 2017 (1980).
- [21] Konstantinos Meichanetzidis, Stefano Gogioso, Giovanni De Felice, Nicolò Chiappori, Alexis Toumi, and Bob Coecke. Quantum natural language processing on near-term quantum computers, 2020.
- [22] G. A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–40, 1995.

- [23] Borja Navarro Colorado. Sistemas de anotación semántica para corpus de español. In Giovanni Parodi, Pascual Cantos, and Lewis Howe, editors, *The Routledge Handbook of Spanish Corpus Linguistics*. Routledge, 2021.
- [24] Arthur Neidlein, Philipp Wiesenbach, and Katja Markert. An analysis of language models for metaphor recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2020.
- [25] L. Padró and E. Stanilovsky. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*. European Language Resources Association, 2012.
- [26] M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics*, 31(1), 2005.
- [27] Andrew Piper. Computational narrative understanding: A big picture analysis. In Yanai Elazar, Allyson Ettinger, Nora Kassner, Sebastian Ruder, and Noah A. Smith, editors, *Proceedings of the Big Picture Workshop*, pages 28–39. Association for Computational Linguistics, 2023.
- [28] John R. Searle. *Actos de habla*. Cátedra, Madrid, 2017 (1965).
- [29] P. Smolensky. The constituent structure of connectionist mental states: A reply to fodor and pylyshyn. In T. Horgan and J. Tienson, editors, *Connectionism and the Philosophy of Mind*. Springer, 1991.
- [30] Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, 2010.
- [31] Dominic Widdows. *Geometry and Meaning*. CSLI Pub., 2004.
- [32] Ludwig Wittgenstein. *Investigaciones filosóficas*. Crítica, 2008 (1953).