Variantes a la extracción automática de vecinos semánticos mediante LSA y al algoritmo de predicación

Guillermo Jorge-Botana
Dpto. de Procesos Cognitivos
Facultad de Psicología
Universidad Complutense de Madrid
jorgeybotana@psi.ucm.es

Ricardo Olmos
Dpto de Psicología Básica
Facultad de Psicología
Universidad Autónoma de Madrid
ricardolmos@inicia.es

José A. León
Dpto de Psicología Básica
Facultad de Psicología
Universidad Autónoma de Madrid
ricardolmos@inicia.es

Francisco Molinero
Dpto. CC. Económicas y Empresariales
Universitat Abat Oliba
fmolineror@uao.es

Resumen

El algoritmo de predicación [2] trata de dar cuenta del proceso que se lleva a cabo en la comprensión de predicaciones. Para ello emplea como base el espacio semántico-vectorial que emana del Análisis de la Semántica Latente LSA. Presentamos en este escrito algunas variantes para la extracción de vecinos semánticos, tanto para términos únicos como para algunas estructuras basadas en predicaciones taxonómicas como "fobia a las tormentas", "fobia a los perros" y para algunos tipos de estructuras metonímicas como "personalidad de la pistola" o "personalidad de los gérmenes" refiriéndose a la personalidad antisocial y a la personalidad obsesiva.

1. Motivación

El Análisis de la Semántica Latente (LSA) devuelve una representación estática del conocimiento representado en un texto. Una matriz LSA representa un conocimiento cristalizado de un dominio temático concreto o de una muestra de uso general del lenguaje. De la anterior proposición se extrae que LSA es únicamente un modelo de representación y adquisición pero no un modelo de procesamiento del lenguaje. La representación proporcionada por LSA funciona como base para proponer modelos de procesamiento e implementar algoritmos que lo simulen *[1]*. Esta base de conocimiento tiene que ser sesgada por un contexto, bien sea éste semántico, sintáctico o de cualquier otra naturaleza de tal manera que simule los procesos observados en el procesamiento de sujetos reales. Tal es el caso por ejemplo del algoritmo de predicación *Kintsch (2001)*.

2. El algoritmo de Predicación

2.1. Marco teórico

El algoritmo de predicación *[2]* pretende resolver algunas limitaciones que posee LSA. Una de las principales desventajas de LSA, aunque para ciertos propósitos no lo sea, es que no posee ningún tipo de análisis sobre las relaciones de orden de las palabras o análisis de los roles que se mantienen dentro de una determinada frase o estructura. Quizás por esta causa, los desarrollos LSA se desvelaban más eficientes al nivel del párrafo, que es donde la intervención del orden es menor o irrelevante *[7,5,6,4]*. Otra limitación es que el cálculo de la suma vectorial para representar estructuras predicativas del tipo "El confidente sopla" suele estar condicionado por como sean las longitudes de los vectores de ambos términos y por lo tanto, será difícil que el vector resultante represente su verdadero sentido. El postulado de partida que mantiene *Kintsch* es que el significado exacto de un predicado depende de los argumentos que le acompañan y que tanto predicado y argumentos, están constreñidos por un orden sintáctico que introduce un sesgo a cada uno de ellos. Tomemos por ejemplo el verbo "soplar":

El borracho sopla.
El alumno sopla.
El viento sopla.
El confidente sopla.

Todas estas frases tienen el verbo soplar cómo denominador común, sin embargo, este mismo verbo adquiere sentidos o significados distintos. Todos sabemos que en "el alumno sopla" el verbo "soplar" no tiene el mismo sentido que en "el viento sopla". El mismo verbo adquiere unas propiedades u otras según sean los argumentos que le acompañan, es decir, las propiedades que dan significado a ese verbo son dependientes del contexto formado por sus argumentos. Supongamos la proposición PREDICADO [ARGUMENTO] expresando qué el predicado devuelve unos valores u otros en función de cuales sean los argumentos(figura 1). Tanto PREDICADO cómo ARGUMENTO estarían representados en sendos vectores. En la manera habitual de LSA, para calcular el vector que representase a la proposición entera, simplemente se hallaría un vector nuevo que fuese la suma o el "centroide" del vector ARGUMENTO y el vector PREDICADO. Por tanto, lo que hace el centroide o suma vectorial en la manera LSA es tomar todas las propiedades del predicado sin discriminar en función de los argumentos y sumarlas a las del argumento. Todas las propiedades del verbo "soplar" serían tenidas en cuenta a la hora de calcular el nuevo vector. Si es el caso de que el argumento tenga una longitud de vector menor que el predicado y que en el predicado, estén mejor representados otros argumentos que no sean el argumento que analizamos, el vector que representa la predicación no captará el verdadero sentido, sino que estarán representadas propiedades de otros argumentos más dominantes.

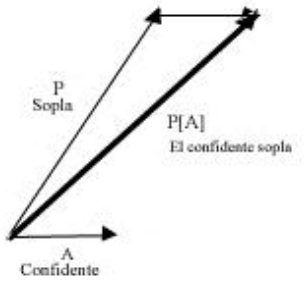

Figura 1. Superioridad del predicado

Como no pasa desapercibido en la figura 1, la solución queda sesgada por la longitud de los vectores. Si como en este caso, el vector del predicado es de mucha mayor longitud que el vector del argumento, el significado de la proposición final no tendrá en cuenta las restricciones que ejercen los argumentos sobre el predicado y tomará las propiedades de sus argumentos más frecuentes. Poner unos argumentos u otros no incidirá apenas en el sentido de las proposiciones. El vector de la proposición estará compuesto por características generales del verbo "soplar" (las propiedades predominantes) sin acotar su significado al contexto que proporciona el argumento "confidente". Probablemente, será el caso de que el vector resultante de "el confidente sopla" estará relacionado con "viento" y con "globo" y no con "crimen", por ejemplo. Para impedir esto, lo que hace el algoritmo de predicación es sesgar el vector-predicado añadiendo un contexto adecuado al tipo de argumento que se está predicando. Este contexto estará formado por los vecinos semánticos del predicado que también estén relacionados con el argumento. Los pasos a seguir podían enumerarse en los siguientes:

1. Se localizan predicado ("soplar") y argumento ("confidente") dentro de una proposición. P(A)
2. Se extraen los n vecinos más próximos al predicado (P). Dado un espacio semántico, se calcularán los cosenos de P con cada uno de los términos que componen el espacio semántico. Hecho esto, se seleccionan los n primeros términos que más se aproximen a P (la elección de n queda abierta al tipo de espacio semántico y observaciones empíricas)
3. Se calculan los coseno entre cada uno de los n términos elegidos y el argumento(A).
4. Se implementa una red en la que se provea de conexiones inhibitorias entre los n términos entre ellos y conexiones activatorias entre cada uno de los n términos con el argumento (A) y con el predicado(P). Las conexiones se harán en base a los cosenos calculados en el paso 2 y 3.
5. Cómo último paso, se calculará el vector P(A) con la suma o centroide del Predicado (p), más el argumento(A), más los términos que reciban más activación en la red implementada, es decir, aquellos que reciban más activación excitatoria de Predicado y argumento y menos inhibición lateral de los propios términos de su misma capa. P(A) será el vector de la predicación.

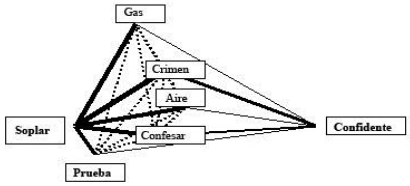

Figura 2. Red de predicación

De esta forma, se impone un sesgo a la manera estándar del centroide para que contemple el fenómeno lingüístico de que el sentido del predicado es dependiente de la información proporcionada por sus argumentos. *Kintsch (2001*) nos muestra el funcionamiento del algoritmo y comprueba el sentido final de algunas predicaciones en lengua inglesa como *"The bridge collapsed", "The plan collapsed", "The runner collapsed" y con un ejemplo que se ajusta más a estructuras taxonómicas o jerárquicas como* "Pelican is a Bird" y "the bird is a pelican" . Además, [2] muestran como el mismo mecanismo de predicación puede servir para modelar la comprensión de la metáfora.

2.2. Implementación :

El esquema general del modelo de predicación puede implementarse de muy diversas maneras y con una gran variedad de metodologías, pero es sin duda el modelo de Programación Orientada a Objetos (POO) el que mejor se adapta a sus singularidades. En general, todos los desarrollos de aplicaciones de redes neuronales artificiales están siendo desarrolladas con dicha metodología pues permiten un poder de abstracción que expresa mejor las entidades de la jerarquía, a saber: Redes, capas, Nodos, Reglas, Protocolos de Entrenamiento, etc. En el caso del algoritmo de predicación, la red que ha de ser diseñada, se ajusta también perfectamente a esta metodología aunque no necesita de todos los componentes que contienen las redes al uso, como reglas de retro-propagación, ciclos de entrenamiento, etc., sino que en un solo ciclo, la propia mecánica de la red

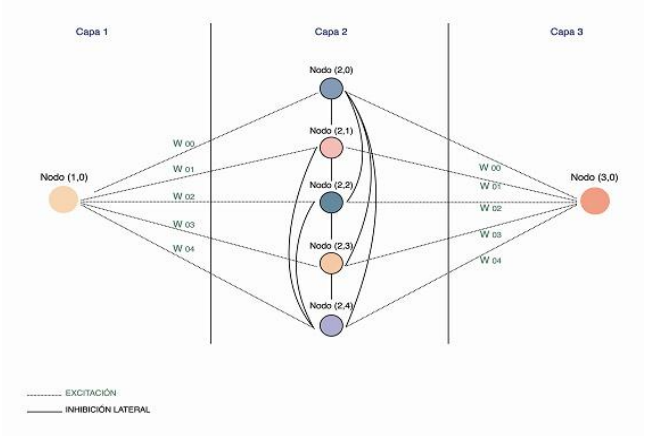

Figura 3. Red , capas y nodos.

con sus propiedades inhibitorias y excitatorias se encarga de equilibrar desde el primer y único ciclo cada uno de los nodos que la componen.

Una red de predicación consta de 3 capas (figura 3), aunque dos de ellas, la primera y la tercera no tengan más que un Nodo. Estos nodos de la primera y tercera capa son los concernientes al *Predicado* (1,0) y al *Argumento* (3,0). La capa central consta de tantos Nodos como vecinos del *Predicado* se deseen contemplar en el algoritmo, aspecto este que está sujeto a factores empíricos. Cada Nodo de esta segunda capa equivale a un término del vecindario semántico del *Predicado*. Además, los Nodos de la capa central, constan de dos mecanismos de activación, el primero es el mecanismo de activación intercapa de forma que cada nodo es activado por los dos miembros de la predicación en la medida en que aumente su índice de similitud con cada uno de ellos. cada uno de los pesos que ponderan la conexión entre predicado y cada uno de los nodos centrales(W00, W01,W02,W04...W0N), estará representado por el coseno entre el predicado y cada nodo central. De la misma forma, las conexiones entre el argumento y cada nodo de la capa central (W00, W10,W20,W40...WN0) serán equivalentes al coseno entre los términos de cada uno de los nodos centrales y dicho argumento. El otro mecanismo es el de inhibición lateral, con el cual, cada nodo recibe inhibición de todos y cada uno de los vecinos de la capa central. De esta forma, cada nodo ejerce de competidor de los demás.

Las conexiones Intercapa de cada predicado con cada nodo de la capa central tienen un peso equivalente al índice de similitud entre los términos que representan cada nodo (esto se expresará generalmente con cosenos). Es decir, Respecto a las conexiones inhibitorias, a cada activación Intercapa de cada nodo central se le restará la suma total de las activaciones de cada uno de los demás nodos en dicha capa. Calculadas las activaciones Inter e Intra, se obtendrá una activación global

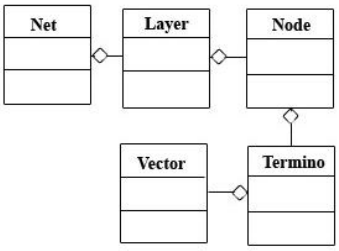

Figura 4. Diagrama de clases.

(Inter – Intra) de cada uno de los nodos de la capa central. Ordenando estos de mayor a menor, se tomarán los n primeros nodos según el índice de activación global. Con los términos que representan estos nodos, más el término predicado y el término argumento, se calculará el centroide, es decir, la suma de todos los vectores de dichos términos y se obtendrá el vector resultante de la predicación(A)

La forma de implementar esta red se puede simplificar en lo siguiente(figura 4):

Tenemos varias entidades que conforman clases:

- *Net:* Es la clase que representa la red entera, esta clases contendrá a su vez objetos de la clase Layer (en el caso del algoritmo de predicación contendrá tres capas) . Es de capital importancia resaltar la existencia de tres procedimientos de clase, uno público y dos privados que se encargarán de conseguir el equilibrio de toda la red.
- *Layer*: Con esta clase se instancian cada una de las capas que se requieren en el algoritmo. A su vez, esta clase contiene Objetos Node. Estos objetos Node configurarán cada una de las capas (objetos Layer) que contendrá la red (Objeto Net). Una particularidad de esta red es que la primera y última capa sólo contendrán un nodo.
- *Node:* Esta configura la unidad más básica en la red y es contenida por los objetos Layer. Aún así, es de capital importancia ya que es aquí donde se contendrá el valor de las activaciones que serán calculados por los procedimientos de la clase Net. Además a la clase *Node* se le puede agregar un objeto Termino construido previamente. La ventaja de agregarle este último objeto es que se podrán utilizar todos sus atributos y métodos para los cálculos de las conexiones, es decir, la clase Término se puede diseñar de manera que tenga métodos para compararse con otro término por medio del coseno, la distancia, etc. De esta manera, los métodos de la clase *Net* que equilibran la red los utilizarán.
- *Termino:* La clase Termino, junto con la Clase Pseudodocumento y alguna más, son clases de uso general en nuestra aplicación LSA, es decir, es la base sobre la que se lleva a cabo el algoritmo de predicación. Lo que interesa es que este objeto representa no sólo un término en formato cadena, sino el vector que lo representa en la matriz, el índice que ocupa en dicha matriz, la longitud de vector, etc. Además, instanciado un Objeto Término, se podrá hacer uso de sus métodos para compararlo con otros Objetos Término y extraer sus n primeros vecinos.
- *Vector*: Esta clase expresa simplemente que un Objeto Termino contiene un Objeto Vector (que puede ser un simple array) que contiene la representación de nuestro literal en el espacio semántico-vectorial.

La secuencia es la siguiente:

1. Se instancia un objeto Net
2. Se instancian Tres objetos Layer
3. Se instancian los nodos(Node) necesarios para rellenar las capas(Layer).
4. Una vez se tienen las tres capas rellenas se procederá a incorporar las capas(Layer) a la Red(Net).
5. A partir de aquí, con todos las capas y nodos incorporados a la red, se procederá a lanzar los cálculos de las activaciones de la red.
6. Se ordenarán los Nodos-Termino de la capa central en base a su activación y se extraerán los n primeros (n es un número empírico que

en nuestro caso es 5). Con los términos de estos 5 nodos más el término Predicado más el término argumento se hará una suma vectorial y el vector resultante será la representación de esa predicación.

3. Espacio semántico de prueba.

Se lleva a cabo LSA sobre un corpus específico de dominio que versa sobre clasificación y descripción de trastornos mentales . Tras las pertinentes purgas el espacio semántico queda definido por 5335 términos insertos en 959 párrafos. Se toma como reducción de dimensiones el criterio del 40% de valor singular acumulado(Wild, 2005) el cual equivale a quedarse con 187 dimensiones. El coseno medio de similitud entre términos es de 0,018 y las DT es 0,074. Tanto LSA como la red de predicación son calculadas con GALLITO, una aplicación basada en LSA implementada con tecnología .NET.

4. Caso I: Estructuras de un solo término.

4.1. Parámetros

Una forma de constatar el sentido o los sentidos que posee una palabra podría ser consignando los vecinos semánticos más próximos a esa palabra y de esa forma, aglutinar todos los términos que se distribuyen en torno a ella. Haciendo esto, obtendríamos el sentido dominante de esa palabra aunque en forma de gradiente, obtendríamos atisbos de otros sentidos minoritarios. La manera de extraer estos vecinos semánticos sería diseñar un procedimiento que vaya calculando el coseno entre ese término y los demás términos del espacio semántico y vaya conservando los n mayores valores en una lista. El resultado final será una lista de los n primeros términos que más similitud guarden con el seleccionado. A su vez y como manera alternativa, se podría también contemplar la posibilidad de que en esta lista, queramos primar vecinos que estén bien representados en el espacio semántico. Habida cuenta de que la longitud de los vectores de un espacio semántico que representa un tema específico muestra cuan bien está representado el conocimiento en términos y documentos *[6]*, proponemos la longitud de vector como factor corrector en la fórmula que compara cada par de términos. En base a estas dos observaciones, se podrían proponer dos formas distintas de extracción de vecinos semánticos.

(1) COSENO
Similitud = Cos(A,I)
(2) COSENO CORREGIDO
Similitud = Cos(A,I) * log (1 + longitudVector(I))

Siendo
I = Los vectores que representan a cada uno de los términos del espacio semántico.
A = El vector que representa al término sobre el cual se quiere extraer la lista de vecinos.

La fórmula (1) es la simple comparación de vectores por medio de los cosenos. En la fórmula (2), se introduce la longitud de vector como factor corrector. De esta manera, en la segunda fórmula, se está promocionando que los términos más representativos del espacio semántico tengan prioridad a la hora de considerarse vecinos.

4.2. Ejemplos.

Los vecinos extraídos con el coseno normal evidencian que el sentido dominante en el término "fobia" es su sentido de "fobia social". Esto se muestra en la frecuencia en que las palabras que forman el vecindario son del ámbito de la fobia social: "social", "público", "timidez", "sociales", "colorado"(figura 5 izquierda)[1], etc y por la ausencia de componentes de otro tipo de fobia o por su aparición en puestos en la parte inferior como "específica". Con el coseno corregido (figura 5 derecha), se promociona que los vecinos más próximos sean por regla general, de corte más generalista por lo que aparecen en los primeros puestos términos como "evitación", "exposición", "específica", "agorafobia", "fobias"

[1] En esta y las siguientes gráficas se ordenan en orden descendente los 21 primeros vecinos extraídos bajo las diferentes formas. Además, en las barras se representan la similitud o coseno con la estructura de referencia(azul) y la longitud de vector de cada vecino, con el fin de constatar la representatividad de los vecinos extraídos.

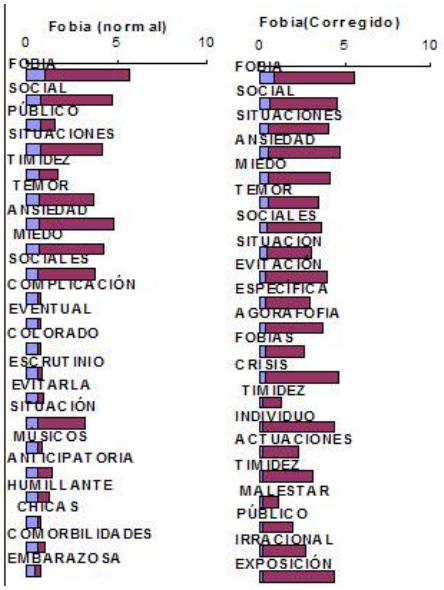

Figura 5. Vecinos de fobia (normal y corregido)

"crisis" y adelantan puestos términos como "situaciones", "miedo", "terror". Por el contrario, , "pierden su estatus términos más concretos como "colorado", "timidez", "humillante", "chicas", "embarazosa", "comercios","reuniones", términos estos más adscritos al sentido dominante de la fobia, la fobia social. A la hora de extraer vecinos con el coseno corregido, lo que se consigue es un vecindario mucho más representativo de los temas que conciernen al término del cual se extraen. La definición que se hace con esta forma de extracción, es de un nivel superior en términos jerárquicos a la que se puede hacer con el coseno normal. Los vecinos serán más conocidos y la definición tendrá un sabor "intensivo" frente al sabor "extensivo" del coseno normal. Esto es más evidente en los vecinos de "tormenta", término este cuya longitud de vector es mucho menor que "fobia". Los vecinos de "tormentas" extraídos con el coseno normal parecen ser términos de su mismo nivel en la definición, es decir, "precipicios", "puentes", "inyecciones", "aviones" ,"serpientes", "recintos", "transportes", "espacios", "atragantamiento", etc. y pueden considerarse como tipos de situaciones a las cuales, los fóbicos tienen miedo. Sin embargo, empleando el coseno corregido, los vecinos que se consiguen son mucho más generales (figura 6 derecha). Nótese también como el propio término "tormentas" no figura entre los primeros puestos dejando paso a términos cuyo sentido es más general.

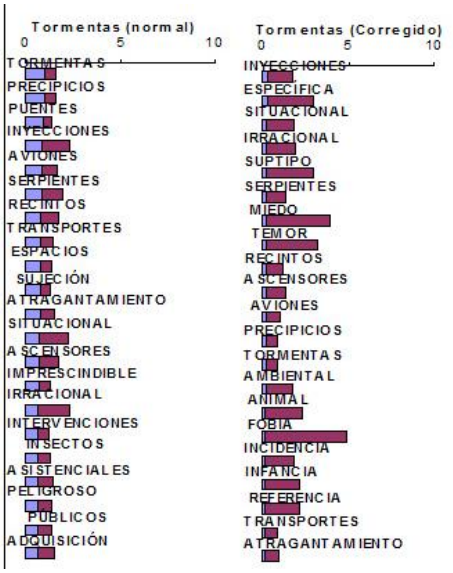

Figura 6. Vecinos de tormentas (normal y corregido)

4.3. Conclusión

Ambas definiciones son claramente diferentes en cuanto a los términos que la definen, la del coseno normal es mucho más concreta, mucho más "extensional" mientras que la del coseno corregido es mucho más abstracta, mucho más "intensional".

De esta forma pueden rastrearse ambos sentidos, la primera forma peinará las relaciones que son eminentemente "entre iguales" y la segunda las relaciones que mantiene con términos de orden superior. Parece observarse en esta primera muestra que contar también con los vecinos extraídos mediante la corrección del coseno, es de mayor utilidad en palabras cuya longitud de vector es pequeña ("tormentas"), aunque puede aportar información a la hora de extraerlos de palabras más representativas del corpus de referencia ("fobia").

5. Caso II: Estructuras predicativas de dos términos (suma vectorial y algoritmo de predicación)

5.1. Parámetros

Existen algunas cuestiones abiertas en torno al cálculo del algoritmo de predicación que están sujetas a observaciones empíricas y que pueden depender del tipo de espacio semántico que se procesa. La primera cuestión es la elección del número de vecinos del predicado que se seleccionarán para configurar la red. Es decir, la elección de los n primeros vecinos del Predicado que participarán en la red. En el caso de *Kintsch [2]*, aunque reconoce que esta magnitud puede ser muy variable dependiendo del grado de relación que tengan predicado y argumento, aconseja que n sea entorno a 20. Esta cifra es aumentada a 500 en la representación de metáforas predicativas pues la relación entre predicado y argumento no es tan estrecha y los términos cruciales que son pertinentes para ambos no se suelen encontrar entre los primeros 100 vecinos del Predicado. En nuestro caso hemos establecido n como el 10% del número total de términos que posee nuestro espacio. Justificamos esta cifra variable como punto de partida por creer que n se encuentra también ligado al tamaño del espacio semántico. Una segunda cuestión es el número de nodos-término activados (k) cuyo vector se tomará para formar la representación final de la predicación P(A). *Kintsch [2]* propone que de 1 a 5 es la cifra que mejor se comporta. Considerar demasiados supondría un riesgo de que nuestra representación se difuminará con valores espurios. Considerar pocos podría conllevar perder información crucial. Nosotros seguimos esta recomendación e igualamos a K el número de 5.Otra cuestión abierta y que nos parece importante es la fórmula de activación de los nodos y en especial la que concierne a la activación intercapa. El valor de activación Inter. de cada nodo puede ser calculada teniendo en cuenta simplemente los cosenos del predicado y del argumento con cada uno de los nodos pero también pueden ser corregida con la manipulación de algunos valores como la longitud de vector de Predicado y Argumento, la desviación típica entre ambos cosenos o la longitud de vector de los vecinos del Predicado que se introducen en la red (*véase esquema general de implementación*).
Siendo:

CosA = Cos(P,i), es decir, el coseno entre el vector del Término Predicado y el Vector de cada Término Nodo(i)
Y
CosB = Cos(i,A), es decir, el coseno entre el Vector de cada Termino Nodo(i) y el vector del Término Argumento.

Las formas más básicas de activación de los nodos serían las que tienen en cuenta el valor del coseno entre cada uno de los vecinos del predicado y su argumento:
Inter. = CosB
En este primer caso, los nodos se activarán más o menos en la medida en que el coseno entre el término representado en ese nodo y el Argumento sea alto. Por lo que nosotros hemos podido comprobar, esta fórmula se suele mostrar más efectiva que la suma del coseno del predicado y el vector nodo más el coseno del vector nodo y el argumento cuya fórmula se expresaría de la siguiente forma.
Inter. = CosA + CosB
Tras sondear algunas posibilidades con los parámetros arriba descritos (longitud de vector de Predicado y Argumento, la desviación típica, etc), proponemos esta fórmula:
*Inter. =CosA + CosB * (1 + log(LongitudVector(Predicado)) +(1/DT(CosA,CosB))*

Aunque la situación más común es que la longitud de vector de un Predicado sea mayor que la del Argumento, puede darse el caso de que la diferencia sea excesiva e impida al propio algoritmo de predicación extraer el verdadero sentido. Para ello, se puede corregir la fórmula multiplicando el coseno entre el Nodo y el argumento por una ponderación basada en la longitud de vector del Predicado. De esta manera, aumentará la participación del argumento en la activación, de manera proporcional a como sea la longitud de vector del predicado. Esto valdrá también para el caso en que el predicado tenga menor longitud de vector pues se disminuirá la participación del argumento.
La corrección basada en la desviación típica se introduce con el motivo de promocionar la activación de términos-nodos cuyos dos cosenos (con el Predicado y el argumento) sean

significativos, es decir, no promocionar los nodos que reciban activaciones unilaterales.
De esta última fórmula extraeremos algunos ejemplo. Para ello, emplearemos dos posibles variantes, a saber, con vecinos corregidos o sin corregir. Al igual que se hacía en la extracción de vecinos de un solo término, una vez calculadas las activaciones de los nodos y extraídos los k primeros nodos más activados para formar el vector de la predicación, se formará la lista de los vecinos modulada o no con la longitud de vector, es decir, promoviendo o no un vecindario más representativo.

5.2. Ejemplos.

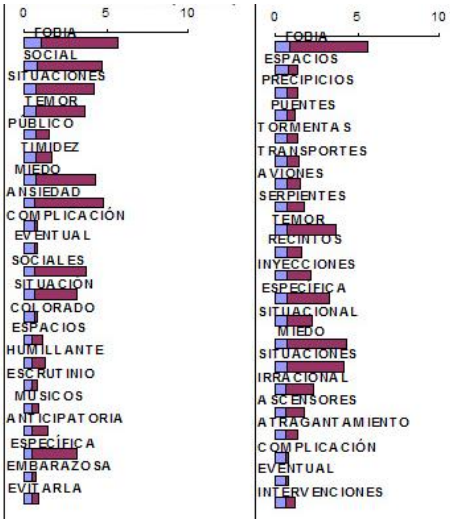

Figura 7. Vecinos “fobia a las tormentas” : sin predicación normal y con predicación normal

En la figura 7 se muestra como el sentido predominante deja su protagonismo para ceder terreno a sentidos más parejos a las fobias específicas: “espacios”, “precipicios”, “puentes”, “serpientes”, “especifica”. En cuanto a la forma de predicación corregida(figura 8 derecha), se observa que los vecinos extraídos tienen longitudes de vector mayores que posibilitan que la definición de la predicación contenga términos más representativos y de niveles jerárquicos más altos aunque esto también permite que esta misma definición siga conteniendo términos del sentido más predominante como “social” o “sociales”.

Parece pues, a la vista de los vecinos extraídos con las formas más eficientes de predicación que el sesgo que se introduce a la hora de representar dicha predicación da cuenta del valor real que esta transmite. En el caso de “fobia a las tormentas”, el término “tormentas introduce unos parámetros que modulan el sentido general y dominante de “fobia”. En este caso, la fobia tiene que poseer unas connotaciones diferentes a las de la fobia “social” pero debe conservar el sentido general y común de las fobias, sentido este que puede ser definido por términos como ‘‘miedo”, temor”, “situacional”, etc. Este sentido general y común resulta más palpable cuando se calcula la representación de la predicación y se extraen sus vecinos atendiendo también a la longitud de vector de os mismos. De esta forma, se consigue que el sentido de la predicación, sea como en los ejemplos de términos únicos, más “extensional” y haga referencia a términos más representativos.

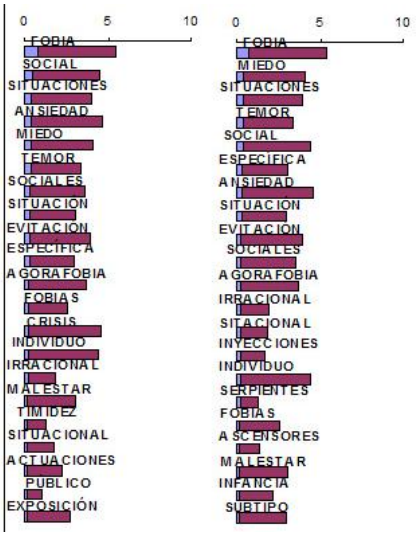

Figura 8. Vecinos “fobia a las tormentas”: sin predicación corregido y con predicación corregido

Siguiendo con los posibles ejemplos, es interesante mostrar un ejemplo que hemos extraído y que empieza a poder interpretarse como un ejemplo más parecido al lenguaje natural que no se ciñe simplemente al ámbito taxonómico.

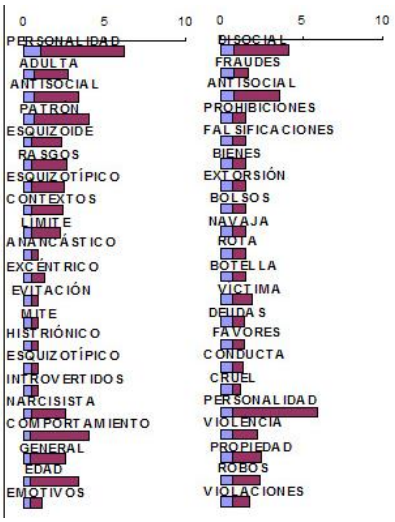

Figura 9. Vecinos "personalidad de la pistola": sin predicación normal y con predicación normal

Imaginémonos que queremos referirnos, en el campo de las patologías psicológicas y psiquiátricas a un patrón de personalidad violento e inadaptado y lo bautizamos metafóricamente como "Personalidad de la pistola". Un interlocutor al que le dijésemos que una persona tiene la "personalidad de la pistola" podría entender a lo que nos referimos si es el caso de que tuviese un modelo mental específico de dominio parecido a este que maneja LSA y podría entenderlo incluso sin haber recibido ninguna especificación al respecto. Este lenguaje pseudo-metafórico puede ser captado con los mismos mecanismos que se emplean para la predicación y es que, abstrayéndonos de ellos, el mecanismo de captación del sentido metafórico de las estructuras sigue respondiendo a la introducción de los sesgos contextuales que un término ejerce sobre otro. En este caso, la palabra "pistola" introduce un sesgo con respecto al sentido amplio de personalidad lo que provoca la activación de los contenidos referentes a un tipo de personalidad específica, en este caso, la personalidad antisocial. Tomando como línea base los vecinos extraídos con la forma normal del coseno, observamos que la estructura "personalidad de la pistola" cobra un sentido mucho más ajustado a la realidad si empleamos el algoritmo de predicación tanto en su versión corregida como en la versión sin corregir. Se puede observar que en la versión normal del coseno(figura 9 izquierda) se intercalan en los primeros puestos términos pertenecientes a otras variedades de trastornos de personalidad como "esquizotípico", "esquizoide", "anancástico", "excéntrico", "introvertidos", etc. Esto parece enderezarse minimamente si empleamos el coseno normal corregido(figura 10 izquierda) auque siguen apareciendo términos como "esquizotípico", "esquizoide", "narcisista". Sin embargo, con ambas versiones de predicación(figura 9 y 10 derecha), parece que el sentido de "personalidad de la pistola" cobra unas connotaciones más afines a su verdadero sentido. En ambas versiones, los términos que aparecen en el listado de vecinos pertenecen a la categoría de"personalidad disocial" que es la más acorde con el sentido real. En la versión no corregida parece observarse que lo vecinos son de un nivel más bajo en cuanto a representatividad de los contenidos ya que aunque se conservan términos de la personalidad general como "Personalidad" y "disocial" aparecen términos del como "fraudes", "prohibiciones", "falsificaciones", "bienes", "extorsión", navaja", etc. Véase que el vecindario del término "pistola" en solitario, se compone de muchos de estos términos de bajo nivel. Nótese también que entre los vecinos de "personalidad" en solitario, sólo se conservan en nuetra predicación los concernientes a la "personalidad disocial " y que estos vecinos de personalidad no se encuentran en el vecindario directo de "pistola" en solitario. Por tanto, ha sido el algoritmo el que ha propiciado esta mixtura de términos procedentes de estos dos términos, con el sesgo introducido por el orden y el rol de sus constituyentes. Esto se comprueba de manera más nítida en la versión de predicación corregida(figura 10 derecha). En esta versión desaparecen los términos como "esquizoide", "esquizotípico", "límite", "narcisista", "evitación" pero se conservan las propiedades generales de la personalidad como "patrón", "comportamiento", "personalidad" además de imponerse otros términos que aún son representativos pues tienen alguna longitud de vector aunque ya son restringidos al ámbito de la personalidad antisocial como "robos", "violencia", "propiedad", "agresivos". Además, también aparecen en las siguientes posiciones términos con

menos longitud de vector como “bienes”, “fraudes”, “daños”, extorsión”, etc.

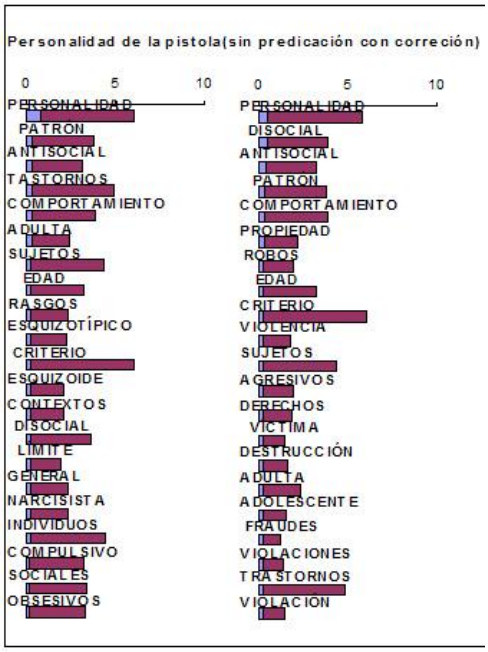

Figura 10. Vecinos “personalidad de la pistola”: sin predicación corregido y con predicación corregido

5.3. Conclusiones.

Las dos formas de predicación (corregida o sin corregir) parecen comportarse bien para extraer el sentido de las estructuras “fobia a las tormentas” y “personalidad de la pistola” respecto a sus líneas bases (vecindario de los términos en solitario o vecindario de las dos palabras en conjunto pero con el coseno normal) además, la forma corregida de predicación parece hacerlo con una mejor definición en la predicación cuyo argumento tiene una menor longitud de vector(“personalidad de la pistola”). En el caso de esta segunda estructura, el algoritmo de predicación parece dar cuenta de un fenómeno habitual en el lenguaje natural, a saber, que un término de un nivel jerárquico mucho menor, designe metonímicamente contenidos de estructuras superiores e iguales.

6. Bibliografía

[1] Burgess.C.(2000)Theory an operational definitions in computational memory models: A response to Glenberg and Robertson. Journal of Memory and Language, 43, 402-408.

[2] Kintsch, W.(2001) Predication. Cognitive Science 25, 173-202

[3] Kintsch, W. and Bowles, A. (2002) Metaphor comprehension:
What makes a metaphor difficult to understand? Metaphor and Symbol, 2002, 17, 249-262"

[4] Kurby, C. A., Wiemer-Hastings, K., Ganduri, N., Magliano, J. P., Millis, K. K., & McNamara, D. S. (2003). Computerizing reading training: Evaluation of a latent semantic analysis space for science text. Behavior Research Methods, Instruments, & Computers, 35, 244-250

[5] Landauer, T. K., Laham, D. & Foltz, P. W. (2003). Automatic Essay Assessment. *Assessment in Education, 10,* 295-308.

[6] Rehder, B., Schreiner, M. E., Wolfe, M. B., Laham, D., Landauer, T. K., & Kintsch, W. (1998). Using Latent Semantic Analysis to assess knowledge: Some technical considerations. *Discourse Processes,* **25**, 337-354.

[7] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In *Artificial Intelligence in Education*, pp. 535–542, IOS Press, Amsterdam, 1999.

[8] Wild. F, Stahl. C , Stermsek. G, Neumann. G : Parameters Driving Effectiveness of Automated Essay Scoring with LSA, in: Proceedings of the 9th International Computer Assisted Assessment Conference (CAA), 485-494, Loughborough, UK, July, 2005