Introducción al Procesamiento de Lenguaje Natural

Junio 2023

Soluciones

Ejercicio 1

i) En cualquier oración, el grupo que cumple la función sintáctica de sujeto respecto al verbo principal tiene siempre el rol semántico agente respecto a este mismo verbo.
***Falso**. Las funciones sintácticas y los roles semánticos no pueden mapearse directamente. Un ejemplo ilustrativo es la voz pasiva: si transformamos una oración como "el perro come huesos" a la voz pasiva, el grupo con rol semántico agente ("el perro"), que en esa oración tiene función sintáctica sujeto, sigue siendo agente en la voz pasiva ("el hueso es comido por el perro") pero su función sintáctica pasa a ser complemento.*

ii) La relación de hiperonimia es simétrica, es decir, si p1 es hiperónimo de p2, entonces p2 es hiperónimo de p1.
***Falso.** La hiperonimia es asimétrica. Si A es hiperónimo de B, entonces B no es hiperónimo de A. La relación inversa de la hiperonimia se llama hiponimia (B sería hipónimo de A). Ejemplo: animal es hiperónimo de perro, pero perro no es hiperónimo de animal (perro es hipónimo de animal)*

iii) El algoritmo de Lesk para desambiguación semántica consiste en elegir el significado cuya glosa comparte más palabras con el contexto de la palabra a desambiguar.
***Verdadero**. El algoritmo asume que la glosa o definición de la palabra probablemente contenga palabras que suelen aparecer en los contextos en donde se usa la palabra a desambiguar. Por ejemplo, si uso la palabra "banco" con el significado "tipo de asiento frecuente en plazas", alguna de esas palabras seguramente aparecerá en contextos en donde se use "banco" con ese significado ("me senté en el banco de la plaza") y no en contextos en donde se use "banco" con otro significado ("fui al banco a retirar plata").*

iv) La morfología derivativa aplicada a los verbos nos permite generar todas sus conjugaciones.
***Falso**. La morfología derivativa combina un afijo con la raíz de una palabra cualquiera, sin importar la categoría, para generar una palabra de otra clase o incluso con otro significado.*

v) Word embeddings es un método de clasificación supervisada que permite predecir, por ejemplo, si un tweet tiene sentimiento positivo o negativo.
***Falso.** Word embeddings es una técnica de representación de palabras, donde a cada palabra le corresponde un vector de valores reales denso. Sí se podría utilizar como entrada para un método de clasificación supervisada, pero no es su única función.*

vi) La Precisión de la clase C mide la cantidad de elementos bien clasificados de esta clase (verdaderos positivos de C) sobre el total de elementos clasificados como clase C (verdaderos positivos de C + falsos positivos de C).
***Verdadero.** Por definición de precisión.*

vii) Un modelo de lenguaje permite obtener la probabilidad de que una secuencia de palabras ocurra en cierto lenguaje.
Verdadero*. En su definición original, un modelo de lenguaje nos permite calcular la probabilidad de la siguiente palabra dadas todas las anteriores en un lenguaje, y por ende la probabilidad de toda la secuencia.*

viii) Las redes neuronales solo pueden utilizarse para clasificación secuencial, tomando N palabras de entrada y devolviendo N elementos de salida.
Falso. *Las redes neuronales pueden utilizarse para problemas de clasificación de distinto tipo: N a 1, N a N, o N a M.*

ix) Clustering es un método de aprendizaje no supervisado.
Verdadero. *Al contrario de clasificación que es supervisado, clustering es una técnica que identifica similitudes entre los objetos y los agrupa según esas características que son comunes.*

x) LDA es un modelo de recuperación de información.
Falso*. Es un modelo probabilístico utilizado en el modelado de tópicos.*

Ejercicio 2

a) ***lento*** es adverbio cuando aparece con el verbo ***camina***, no hay concordancia con el sujeto porque los adverbios no tienen flexión de número ni género. ***lento*** y ***lenta*** son adjetivos cuando aparecen con el verbo ***es***, en esos casos hay concordancia con el nombre que funciona como sujeto del verbo.

b)

o → gn gv
gn → npropio (*)
gv → v adv | v adj
npropio → María | Pedro
v → camina | es
adv → lento
adj → lento| lenta

c) Tomando en cuenta la gramática generada en b), antes de aplicar CKY es necesario eliminar la producción unitaria gn → npropio porque pasó a ser una variable no alcanzable y una G en FNC debe estar simplificada.

Queda entonces la siguiente gramática en FNC

o → gn gv
gn → María | Pedro
gv → v adv | v adj
v → camina | es
adv → lento
adj → lento| lenta

Aplicamos CKY

	Maria (1)	camina (2)	lento (3)
0	gn		o
1		v	gv
2			adj adv

Con lo cual la oración es posible de generar.

d) La gramática, como suele suceder con las GLC, no permite chequear la concordancia. Para resolverlo se podrían escribir reglas más específicas, que distinguieran los verbos que admiten adjetivos y los que admiten adverbios, por un lado, y escribir reglas para npropio masculinos y npropio femeninos, además de adjetivos masculinos y adjetivos femeninos. También pueden usarse otros formalismos gramaticales, como HPSG, que incorpora rasgos para chequear concordancia.

Ejercicio 3

a) |V| = 9

p(<S> María come pan </S>) =

= p(<S> María) p(María come) p(come pan) p(pan </S>) =

= 2/11 2/11 2/11 2/10 =

= 0.00120

p(<S> María comerá pan </S>) =

= p(<S> María) p(María comerá) p(comerá pan) p(pan </S>) =

= 2/11 2/11 1/10 2/10 =

= 0.00066

Según este modelo, "María come pan" es más probable que "María comerá pan".

b) O → GV GN / O → GN GV

GN → Det N GP / GN → GP GN

GP → Prep Det N / GP → N

Ejercicio 4

a) Utilizando el algoritmo de cálculo de la distancia de Levensthein entre dos palabras, calcule la distancia entre CAMINO y CASIN indicando luego el camino utilizado para calcularla.

	#	C	A	S	I	N
#	**0**	1	2	3	4	5
C	1	**0**	1	2	3	4
A	2	1	**0**	1	2	3
M	3	2	1	**2**	3	4
I	4	3	2	3	**2**	3
N	5	4	3	4	3	**2**
O	6	5	4	5	4	**3**

La DME entre esas palabras es 3.

En "negrita" se marca el camino realizado para su cálculo

b) En un sistema de recuperación de información basado en el modelo vectorial se pretende asignar pesos a un documento A en base a 4 términos: banco, plaza, fuente, economía.

Se pretende seguir un modelo tf-idf para la asignación de pesos. Se pide:

i) Defina en qué consiste el modelo tf-idf.

tf-idf es un modelo que sirve – entre varios usos – para la asignación de pesos en un modelo vectorial de recuperación de información. El peso de un término en un documento aumenta si este aparece mucho en un documento y disminuye si aparece mucho en todos los demás. Para cada término en un documento se define como combinación de la frecuencia de aparición del término – cantidad de veces que aparece el término en la colección - (tf), y la frecuencia inversa del documento (idf); esto significa que cuanto menor sea la cantidad de documentos, así como la frecuencia absoluta de aparición del término, mayor será su factor *idf* y a la inversa.

Además, el *tf* se suele normalizar para ser más justos en relación a los documentos más largos; de donde f_{ij} = cantidad de ocurrencias del término *i* en el documento *j* entonces $tf_{ij} = f_{ij} / max \{f_{ij}\}$

ii) Se tiene para cada término en el documento A la siguiente frecuencia de aparición de cada uno: banco (4) – plaza (5) – fuente (2) - economía (3)

y que la frecuencia inversa de cada uno es: banco (2) – plaza (3) – fuente (3) - economía (4).

Calcule el vector de pesos del documento A.

El cálculo del vector de pesos correspondiente al documento A se calcularía entonces realizando para cada término $w_{ij} = tf_{ij} * idf_i$ de donde:

banco: tf = 4/5 = 0.8 idf = 2 ⇒ 0.8 * 2 = 1.6

plaza: tf = 5/5 = 1 idf = 3 ⇒ 1 * 3 = 3

fuente: tf = 2/5 = 0.2 idf = 3 ⇒ 0.4 * 3 = 1.2

economía: tf = 3/5 = 0.6 idf = 4 ⇒ 0.6 * 4 = 2.4

de donde el vector del documento A es (1.6, 3 , 1.2 , 2.4)