Introducción al Procesamiento de Lenguaje Natural

Diciembre de 2016

Consideraciones generales
i)La prueba es sin material escrito.
ii)Escriba nombre y C.I. en todas las hojas.
iii)Numere todas las hojas.
iv)En la primera hoja, indique el total de hojas.
v)Comience cada ejercicio en una hoja nueva.
vi)Utilice las hojas de un solo lado.
vii)Entregue los ejercicios en orden
viii)El total de puntos es 40

Ejercicio 1 [8 puntos]

i) ¿Qué es un corpus paralelo? ¿Para qué se utiliza?

Un corpus paralelo es una colección de textos en dos idiomas, donde los textos en un idioma se corresponden con los textos en el otro (son traducciones del mismo texto en dos idiomas). Se utilizan en el contexto de la traducción automática estadística para construir un modelo de traducción entre los dos idiomas. A partir de los textos del corpus se aprende la frecuencia de traducción de pares de palabras o frases.

ii) Indique cuáles son los tres niveles de alineación que puede tener un corpus paralelo. ¿Cuál de estos niveles de alineación es más difícil de conseguir?

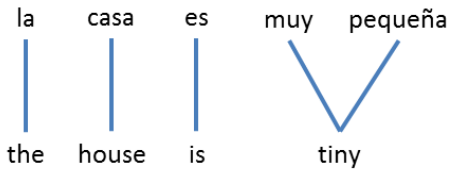
Los corpus paralelos pueden estar alineados en estos niveles:

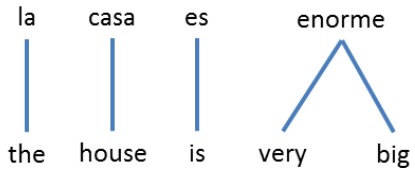
- *A nivel de documento: se conoce cuáles documentos del idioma A se corresponden con los documentos del idioma B.*
- *A nivel de oración: dentro de cada documento, se conoce cuál oración en el idioma A es traducción de otra oración en otro idioma B.*
- *A nivel de palabra: se conoce además cuáles palabras dentro de una oración en el idioma A se traducen como las palabras de otra oración en el idioma B.*

Las alineaciones entre palabras pueden ser de muchos-a-muchos. En la práctica es muy difícil encontrar un corpus que esté alineado a nivel de palabras. Es más sencillo confeccionar un corpus alineado a nivel de documento y utilizar un algoritmo simple (por ejemplo Gale-Church) para alinearlo a nivel de oración. El proceso para alinear el corpus a nivel de palabra en general se realiza en conjunto con la construcción de un modelo de traducción y es un subproducto del entrenamiento.

iii) Dibuje las alineaciones a nivel de palabra entre los siguientes pares de oraciones. ¿Qué problema presenta el uso de funciones de alineación para representar las alineaciones de estas oraciones?

1. a) *La casa es muy pequeña*
   b) *The house is tiny*
2. a) *La casa es enorme*
   b) The house is very big





Si queremos representar los dos pares de oraciones mediante funciones de alineación al mismo tiempo (por ejemplo para confeccionar un corpus alineado a nivel de palabras), tendremos el problema de que una función de alineación solo permite representar relaciones de uno-a-muchos, pero no de muchos-a-uno.
Si elegimos representar en la dirección español->inglés, tenemos que la primera oración se puede escribir como la función a: [1,2,3,4,4], pero no podemos representar la segunda oración porque la última palabra en español tendría dos correspondientes. Un problema análogo ocurre si tratamos de representar la dirección inglés->español.

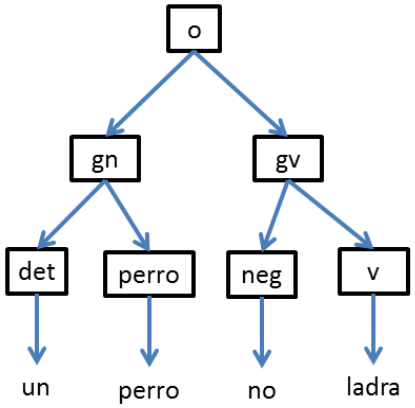
Ejercicio 2 [10 puntos]

i) Por ejemplo: Word embeddings

Los word embeddings son representaciones de un conjunto de palabras como un vector de valores reales, con la particularidad de que la dimensión de los vectores reales es mucho menor que el tamaño del conjunto de palabras. Por ejemplo, se representan miles de palabras del idioma en vectores de 100, 200 o 300 valores reales. Existen diferentes técnicas para entrenar word embeddings (como word2vec y GloVe), y siempre se entrenan a partir de texto no etiquetado, por lo que es relativamente fácil obtener grandes cantidades de palabras para el entrenamiento.

La característica interesante de los vectores de palabras entrenados mediante word embeddings es que palabras semánticamente similares tendrán asociados vectores que estarán cerca en el espacio, mientras que palabras no relacionadas estarán más lejos. De esta manera podemos calcular la similitud entre dos palabras como la distancia (por ejemplo euclídea) entre los vectores que representan dichas palabras.

ii)



o.sem
(aplicación de reglas)
= gn.sem(gv.sem)
(aplicación de reglas)
= det.sem(n.sem)(neg.sem(gv.sem))
(aplicación de reglas)
= (λ P . λ Q . ∃ x P(x) ∧ Q(x))(λ x . perro(x))((λ P .λ x . ¬ P(x))(v.sem))
(aplicación de reglas)
= (λ P . λ Q . ∃ x P(x) ∧ Q(x))(λ x . perro(x))((λ P .λ x . ¬ P(x))(λ x . ladra(x)))
(cambio de variable)
= (λ P . λ Q . ∃ x P(x) ∧ Q(x))(λ y . perro(y))((λ P .λ x . ¬ P(x))(λ x . ladra(x)))
(aplicación funcional)
= (λ Q . ∃ x (λ y . perro(y))(x) ∧ Q(x))((λ P .λ x . ¬ P(x))(λ x . ladra(x)))
(aplicación funcional)
= (λ Q . ∃ x perro(x) ∧ Q(x))((λ P .λ x . ¬ P(x))(λ x . ladra(x)))
(cambio de variable)
= (λ Q . ∃ x perro(x) ∧ Q(x))((λ P .λ x . ¬ P(x))(λ y . ladra(y)))
(aplicación funcional)
= (λ Q . ∃ x perro(x) ∧ Q(x))(λ x . ¬ (λ y . ladra(y)(x))
(aplicación funcional)
= (λ Q . ∃ x perro(x) ∧ Q(x))(λ x . ¬ ladra(x))
(cambio de variable)
= (λ Q . ∃ x perro(x) ∧ Q(x))(λ y . ¬ ladra(y))
(aplicación funcional)
= ∃ x perro(x) ∧ (λ y . ¬ ladra(y))(x)
(aplicación funcional)
= ∃ x perro(x) ∧ ¬ ladra(x)

Ejercicio 3

Considere la siguiente gramática, similar a la vista en el curso:

O	→	GV | GN GV
GN	→	Det Nom | Nom | GN GP
GV	→	V | V GN | V GN GP
GP	→	Prep GN
Det	→	una | un | el | la
Nom	→	Juan | sopa | leche | perro | lente | lentes | sal | tomo
V	→	tomo | toma | sal |
Prep	→	con | sin

1. Aplique el algoritmo CKY para la entrada "*un perro con lentes toma sopa sin sal*", usando la gramática G. ¿Qué salida devuelve el algoritmo?

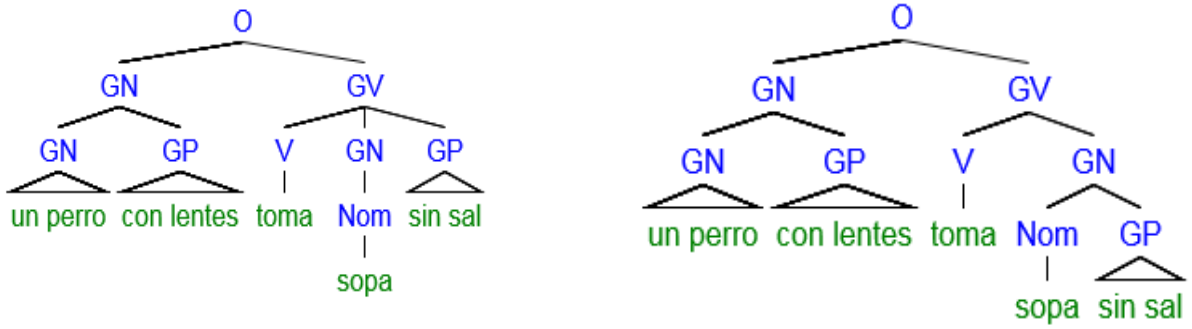
Llevamos la gramática a Forma Normal de Chomsky

O	→	*tomo | toma | sal | V GN | Aux1 GP | GN GV*
GN	→	*Det Nom | Juan | sopa | leche | perro | lente | lentes | sal | tomo | GN GP*
GV	→	*tomo | toma | sal | V GN | Aux1 GP*
GP	→	*Prep GN*
Det	→	*una | un | el | la*
Nom	→	*Juan | sopa | leche | perro | lente | lentes | sal | tomo*
V	→	*tomo | toma | sal*
Prep	→	*con | sin*
Aux1	→	*V GN*

	un (1)	*perro (2)*	*con (3)*	*lentes (4)*	*toma (5)*	*sopa (6)*	*sin (7)*	*sal (8)*
0	*Det*	*GN*		*GN*	*O*	*O*		*O*
1		*Nom, GN*		*GN*	*O*	*O*		*O*
2			*Prep*	*GP*				
3				*GN, Nom*	*O*	*O*		*O*
4					*O, GV, V*	*O, GV, Aux1*		*O,GV,Aux1*
5						*GN, Nom*		*GN*
6							*Prep*	*GP*
7								*O, GN,GV,Nom,V*
8								

El algoritmo devuelve True, indicando que la secuencia presentada puede derivarse a partir de la gramática.

2. Dibuje los árboles sintácticos posibles para la oración.



[O [GN [^GN un perro] [^GP con lentes]] [GV [V toma] [GN [Nom sopa]] [^GP sin sal]]
[O [GN [^GN un perro] [^GP con lentes]] [GV [V toma] [GN [Nom sopa] [^GP sin sal]]]

3. ¿Es posible obtener, a partir de la aplicación del algoritmo, todas las subsecuencias de palabras de la entrada que tienen estructura de oración? En caso afirmativo, lístelas.

Sí, es posible. Basta buscar las ocurrencias del símbolo O en el chart. En el ejemplo, las subsecuencias son:

un perro con lentes toma
un perro con lentes toma sopa
un perro con lentes toma sopa sin sal
perro con lentes toma
perro con lentes toma sopa
perro con lentes toma sopa sin sal
lentes toma
lentes toma sopa
lentes toma sopa sin sal
toma
toma sopa
toma sopa sin sal
sal

4. Mencione una oración generable por la gramática que no sea sintácticamente válida en el idioma español. ¿Cómo se podría modificar la gramática o el formalismo para enfrentar estos casos?

   "La perro toma el sopa" es sintácticamente inválida (falla la concordancia entre determinante y nombre). Esto podría resolverse definiendo reglas diferentes para las cada combinación género/número.

Ejercicio 4 [4 puntos]

Los adjetivos en el idioma fictio Itio están compuestos por una raíz (por ejemplo: "xax", que quiere decir "pálido", y "ses", que quiere decir "ligeramente desviado hacia la derecha"), un sufijo opcional "a" (indicando género femenino), y otro sufijo (obligatorio), que vale "x" si el adjetivo es usado en forma positiva, y "v" si es usado en forma negativa. Por cuestiones ortográficas, en Itio no está permitida la duplicación de la letra "x". Algunos ejemplos de palabras y sus correspondientes análisis:

xaxax	xax+Fem+Pos
xax	xax+Masc+Pos
xaxv	xax+Masc+Neg
sesav	ses+Fem+Neg

Implemente un analizador para los adjetivos de Itio utilizando el álgebra de expresiones regulares vista en clase.

Solución:

```
define root [{xax}|{ses}] %+Adj:0;
define genero [%+Fem:a | %+Masc:0];
define polaridad [%+Pos:x | %+Neg:v];
```

```
define ortografia xx -> x;
define adjitio [root genero polaridad] .o. ortografia;
```

Ejercicio 5 [4 puntos]

Considere un corpus de entrenamiento conformado por los siguientes documentos:

Texto	**Clase**
un gran juego	Deporte
un juego limpio pero olvidable	Deporte
la elección finalizó	No deporte
muy limpio partido	Deporte
fue una elección limpia	No deporte

Utilice un clasificador Naïve Bayes con un modelo Bag of Words (e incorporando suavizado) para estimar la clase del documento cuyo texto es: "el juego finalizó temprano". Justifique sus resultados.

Calculamos primero las probabilidades a priori de cada clase (utilizando máxima verosimilitud):

P(Deporte)=3/5
P(No deporte)=2/5

Luego estimamos las probabilidades de cada palabra según la categoría

P("El" | Deporte) = (0 + 1)/(11+14) P("Juego"| Deporte) = (2 + 1)/(11+14) P("finalizó" | Deporte)=(0 + 1)/(11+14) P("temprano"| Deporte)=(0 + 1)/(11+14)	P("El" | No Deporte) = (0 + 1)/(7+14) P("Juego"| No Deporte) = (0 + 1)/(7+14) P("finalizó" | No Deporte)=(1 + 1)/(7+14) P("temprano"| No Deporte)=(0 + 1)/(7+14)

Entonces, utilizando NB:

argmax C P(C | el juego finalizó temprano) =argmax P(el juego finalizó temprano | C) P(C)

P(el juego finalizó temprano | Deporte) P(Deporte) = P(el|Deporte)P(juego|Deporte)P(finalizó|Deporte)P(temprano| Deporte)= 4,61 x 10^{-6}

P(el juego finalizó temprano | No Deporte) P(No Deporte) = P(el|NoDeporte)P(juego|NoDeporte)P(finalizó| NoDeporte)P(temprano|NoDeporte)= 4,11 x 10^{-6}

La clase más probable, y por lo tanto la que asignaremos al documento es "Deporte"