Introducción al Procesamiento de Lenguaje Natural

Diciembre de 2013

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

Mencione al menos tres cualidades que deberían encontrarse en un sistema de recuperación de información y explique brevemente en qué consisten.

Solución:

Precisión: porcentaje de información correcta respecto a la información total que contiene el sistema. Una precisión baja, lleva a una falta de credibilidad de un sistema

Oportunidad: tiempo transcurrido desde el momento en que se generó el hecho que originó la información hasta el momento en que ésta se pone a disposición del usuario

Completitud: la información ha de ser completa (o al menos lo más posible) para que el sistema sea útil. Depende básicamente de 2 factores: los datos existentes en el sistema y aquellos que se pueden recuperar a partir de una consulta.

Significado: la información debe poseer la mayor cantidad de contenido semántico posible.

Integridad: la información debe ser coherente en si misma y consistente con las reglas semánticas propias del mundo real que representa .

Seguridad: protección de la información frente al deterioro físico o lógico y frente a accesos no autorizados .

Ejercicio 2

¿Para qué se utiliza la métrica BLEU? Escriba la fórmula y explique cómo se calcula dicha métrica.

Solución:

La métrica BLEU se utiliza para evaluar la calidad de la traducción automática. Su fórmula es:

$$BLEU = BP \exp\left(\sum_{1}^{N} w_n \log p_n\right)$$

Para calcularla se debe considerar las oraciones de referencia (correctas) y las oraciones candidatas (salida del sistema que estamos evaluando).
- Los términos p_n son la "precisión de orden n", o sea la cantidad de n-gramas presentes en el candidato que también están en la referencia, dividido por la cantidad de n-gramas en la referencia.
- Los términos w_n son pesos para darle mayor o menor relevancia a los n-gramas de cierto orden.
- BP es la penalización por brevedad, que se usa para penalizar traducciones candidatas muy cortas. Si R' es el largo de todas las oraciones de referencia, y C' es el largo de todas las candidatas, BP se calcula como:

Ejercicio 3

Considere la siguiente gramática con anotaciones semánticas:

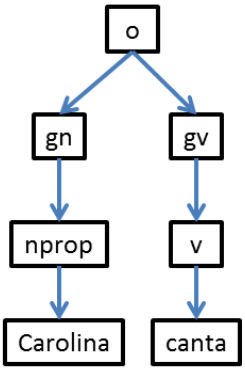
o → gn gv	o.sem = gn.sem(gv.sem)
gn → nprop	gn.sem = nprop.sem
gn → det nom	gn.sem = det.sem(nom.sem)
nom → n adj	nom.sem = adj.sem(n.sem)
nom → n	nom.sem = n.sem
gv → v	gv.sem = v.sem
nprop → Carolina	nprop.sem = λ P . P(carolina)
n → gato	n.sem = λx . gato(x)
v → salta	v.sem = λx . salta(x)
v → canta	v.sem = λx . canta(x)
adj → blanco	adj.sem = λP . λx . P(x) Λ blanco(x)
det → todo	det.sem = λP . λQ . ∀x P(x) → Q(x)
det → un	det.sem = λP . λQ . ∃x P(x) → Q(x)

Dibuje el árbol sintáctico y derive la expresión lógica asociada a las siguientes oraciones:

1. *Carolina canta.*
2. *Un gato blanco salta.*

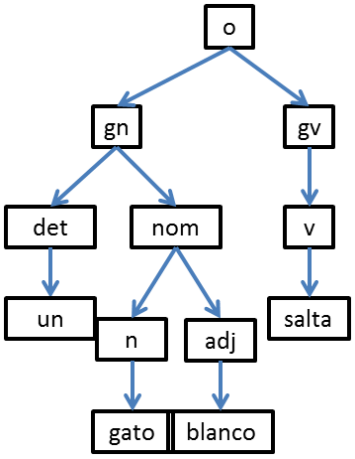
Solución:

1. *Carolina canta.*



o.sem =
[sustitución según reglas]
= gn.sem(gv.sem) =
[sustitución según reglas]
= nprop.sem(v.sem) =
[sustitución según reglas]
= (λP . P(carolina)) (λx . canta(x)) =
[aplicación funcional λP por (λx . canta(x))]
= (λx . canta(x)) (carolina) =
[aplicación funcional λx por carolina]
= canta(carolina)

2. *Un gato blanco salta.*



o.sem =
[sustitución según reglas]
= gn.sem(gv.sem) =
[sustitución según reglas]
= (det.sem(nom.sem)) (v.sem) =
[sustitución según reglas]
= ((λP . λQ . ∃ x P(x)→Q(x))(adj.sem(n.sem))) (λx . salta(x)) =
[sustitución según reglas]
= ((λP . λQ . ∃ x P(x)→Q(x))((λP . λx . P(x) Λ blanco(x))(λx . gato(x)))) (λx . salta(x)) =
[cambio de variable]
= ((λP . λQ . ∃ x P(x)→Q(x))((λP . λx . P(x) Λ blanco(x))(λy . gato(y)))) (λx . salta(x)) =
[aplicación funcional λP por (λy . gato(y))]
= ((λP . λQ . ∃ x P(x)→Q(x))(λx . (λy . gato(y))(x) Λ blanco(x))) (λx . salta(x)) =
[aplicación funcional λy por x]
= ((λP . λQ . ∃ x P(x)→Q(x))(λx . gato(x) Λ blanco(x))) (λx . salta(x)) =
[cambio de variable]
= ((λP . λQ . ∃ x P(x)→Q(x))(λy . gato(y) Λ blanco(y))) (λx . salta(x)) =
[aplicación funcional λP por (λy . gato(y) Λ blanco(y))]
= (λQ . ∃ x (λy . gato(y) Λ blanco(y))(x)→Q(x)) (λx . salta(x)) =
[aplicación funcional λy por x]
= (λQ . ∃ x gato(x) Λ blanco(x)→Q(x)) (λx . salta(x)) =
[cambio de variable]
= (λQ . ∃ x gato(x) Λ blanco(x)→Q(x)) (λy . salta(y)) =
[aplicación funcional λQ por (λy . salta(y))]
= ∃ x gato(x) Λ blanco(x)→(λy . salta(y))(x) =
[aplicación funcional λy por x]
= ∃ x gato(x) Λ blanco(x)→salta(x)

Ejercicio 4

Considere la siguiente gramática:

```
O       →    GN GV
GN      →    Nom | Det Nom | Nom GP
GV      →    V | V GN | GV GP
GP      →    Prep GN
Det     →    el | la | los | las
Nom     →    Juan | cocina | perro
V       →    cocina | cocinan | come | comen
Prep      →    en | a
```

a) Aplique el algoritmo Earley a la oración *Juan cocina en la cocina,* e indique si la oración puede ser generada por la gramática (dibujando en tal caso el árbol sintáctico de la oración).

b) La gramática G reconoce oraciones agramaticales en las cuales falla la concordancia en número o género. Dé dos ejemplos de estas oraciones. ¿Es posible reescribir G de modo que no las reconozca? ¿Qué ventajas tendría una gramática HPSG sobre la gramática libre de contexto G, en cuanto al chequeo de concordancia en número y género?

Solución

Parte a)

0 Juan 1 cocina 2 en 3 la 4 cocina 5

```
chart[0]

dummy  -> .O              [0,0]         Dummy
O      -> . GN GV         [0,0]         Predict
GN     -> . Nom           [0,0]         Predict
GN     -> . Nom GP        [0,0]         Predict
GN     -> . Det Nom       [0,0]         Predict

chart[1]

Nom    -> Juan .          [0,1]         Scanner
GN     -> Nom .           [0,1]         Complete
GN     -> Nom . GP        [0,1]         Complete
O      -> GN . GV         [0,1]         Complete
GP     -> . Prep GN       [1,1]         Predict
GV     -> . V             [1,1]         Predict
GV     -> . V GN          [1,1]         Predict
GV     -> . GV GP         [1,1]         Precict

chart[2]

V      -> cocina .        [1,2]         Scanner
GV     -> V .             [1,2]         Complete
GV     -> V . GN          [1,2]         Complete
O      -> GN GV .         [0,2]         Complete
GV     -> GV . GP         [1,2]         Complete
GN     -> . Nom           [2,2]         Predict
GN     -> . Nom GP        [2,2]         Predict
GN     -> . Det Nom       [2,2]         Predict
GP     -> . Prep GN       [2,2]         Predict

chart[3]
```

```
Prep    -> en .             [2,3]          Scanner
GP      -> Prep . GN        [2,3]          Complete
GN      -> . Nom            [3,3]          Predict
GN      -> . Nom GP         [3,3]          Predict
GN      -> . Det Nom        [3,3]          Predict


chart[4]
Det     -> la .             [4,4]       Scanner
GN      -> Det . Nom        [3,4]       Complete

chart[5]
Nom     -> cocina .         [5,5]       Scanner
GN      -> Det Nom .          [3,5]       Complete
GP      -> Prep GN .        [2,5]       Complete
GV      -> GV GP .          [1,5]       Complete
O       -> GN GV .          [0,5]       Complete
dummy   -> O .              [0,5]       Complete
```

La oración puede ser generada por la gramática. El árbol sintáctico construido es el siguiente:

```
O  -> GN  -> Nom -> Juan
   -> GV  -> GV  -> V -> cocina
          -> GP  -> Prep -> en
                 -> GN  -> Det -> la
                        -> Nom -> cocina
```

Parte b)

Ejemplos de oraciones con problemas de concordancia en género o número reconocidas por G:

* Juan cocina en los cocina.
(El problema está al interior del GN que es complemento del verbo, no hay concordancia ni en género ni en número entre determinante y nombre.)

* El perro comen.
(El problema está en la concordancia entre el GN sujeto y el verbo, el número no concuerda.)

Podemos resolver estos problemas con una GLC especificando las reglas léxicas de modo de tener símbolos distintos para los diferentes nombres y determinantes según su género y número y símbolos distintos para los verbos según su número:

NSM → perro, Juan
NSF → cocina
NPM → perros
NPF → cocinas
DSM → el
DSF → la
DPM → los
DPF → las
VS → come | cocina
VP → comen | cocinan

Las reglas que forman grupos y las reglas para la oración también deben ser especificadas de modo de garantizar la concordancia entre determinante y nombre, por un lado, y entre el GN sujeto y el verbo, por otro:

O → GNS GVS | GNP GVP
GNS → NSM | NSF | DSM NSM | DSF NSF | NSM GP | NSF GP
GNP → NPM | NPF | DPM NPM | DPF NPF | NPM GP | NPF GP

GVS → VS | VS GN | GVS GP
GVP → VP | VP GN | GVP GP

En una gramática HPSG podemos definir rasgos que representen el género y el número para cada categoría. No necesitamos, por lo tanto, subdividir cada categoría en subcategorías (como nombre singular masculino, nombre singular femenino, verbo singular, etc.). Esto nos permite mantener las categorías nombre y verbo, sin perder generalidad ni apropiación desde el punto de vista lingüístico. Los rasgos nos permiten, para cada entrada léxica, especificar los valores apropiados de género y número.

Ejercicio 5

Calcule la distancia de edición entre CALMA y CLAMAR (utilizando distancia de Levenhstein, es decir que todas las operaciones de edición valen 1), y muestre una lista de ediciones posible con esa distancia. Justifique.

Solución

R	6	5	4	4	4	**3**
A	5	4	3	3	3	**2**
M	4	3	2	2	**2**	3
A	3	2	1	**2**	2	2
L	2	1	**1**	1	2	3
C	1	**0**	1	2	3	4
#	**0**	1	2	3	4	5
	#	**C**	**A**	**L**	**M**	**A**

Distancia de edición mínima: 3

Lista de ediciones

Mantener la C
Cambiar A por L (Costo=1)
Cambiar L por A (Costo=1)
Mantener la M
Mantener la A
Insertar una R (Costo =1)

Ejercicio 6

Se tiene un método de clasificación (que incluye un parámetro α que puede valer 1, 0.1 o 0), y se quiere utilizarlo para determinar si una oración dada es un insulto o no. Se cuenta con un corpus de 10.000 oraciones, anotada cada una de ellas con la clasificación esperada. Indique brevemente cuáles serían los pasos generales para construir un clasificador y evaluar su performance. **No** se pide que especifique cuáles serían los atributos utilizados.

Solución

1.Separar el corpus en entrenamiento (80%) y evaluación (20%)
2. Separar un 10% del corpus de entrenamiento para ajuste de parámetros
3. Definir los atributos
3. Entrenar sobre el 90% del corpus de entrenamiento, utilizando los tres valores de los parámetros. Elegir el que obtenga mejor performance en el corpus de ajuste
4. Entrenar sobre todo el corpus de entrenamiento
5. Evaluar el clasificador en el corpus de evaluación. Reportar Precisión, Recall y medida F.