I. Romina Altamirano

Famaf
Universidad Nacional de Córdoba
Argentina

IRASubcat: un sistema para adquisición automática de marcos de subcategorización a partir de corpus

Contenidos

- Subcategorización verbal
- Input XML – Output XML - Configuración
- Aplicación al español (corpus Sensem)
- Aplicación al ruso (corpus plano)

Qué es la subcategorización verbal

“El nene juega a la pelota”

[El nene a la pelota]

el nene jugar a el pelota

ART NC Vlfin PREP ART NC

“Las nenas juegan a las muñecas”

ART NC Vlfin PREP ART NC

Qué es la subcategorización verbal

El marco de subcategorización para un verbo particular, es el conjunto de **patrones con los que el verbo puede ocurrir** en oraciones bien formadas gramaticalmente.

Los niveles relevantes a un lexicón verbal conciernen todos los aspectos léxicos desde morfológicos a sintácticos y semánticos.

Interés de la Subcategorización en PLN

- Analizadores sintácticos
- Traducción automática
- Respuesta a preguntas
- Identificación de oraciones agramaticales

Liliana compra frutas
NP Vlfin NC
Liliana camina frutas
NP Vlfin NC

Lexicones de subcategorización

Creación manual:

- costosa,
- requiere mucho esfuerzo,
- raramente completa,
- no se actualiza fácilmente, y
- sin información de frecuencias.

Adquisición automática:

- disponibilidad de corpus en internet
- velocidad de las computadoras

Identificar marcos y asociar marcos a verbos

- Identificar el verbo
- Algunas palabras o características en un ámbito particular, serán los componentes del patrón.
- En distintos niveles de abstracción los patrones tienden a repetirse...
- Es necesario filtrar la salida
  - Test de frecuencia
  - Test de hipótesis

Input del sistema

- XML – UTF-8 – Muy fácil modificarlo para tomar UTF-16
- Corpus a diferentes niveles de etiquetado

Configuración

- Altamente parametrizable
  - Considerar sólo un conjunto de verbos
  - Actualizar un diccionario existente
  - Longitud del patrón considerado
  - Completar el patrón con una palabra comodín
  - Tener en cuenta el orden, marcar posición del verbo
  - Características lingüísticas que se considerarán
  - Frecuencia mínima para considerar un verbo
  - Frecuencia mínima para considerar una coocurrencia
  - Si se usará test de hipótesis likelihood ratio
  - Si se colapsarán patrones (identificando opcionales)

Output

- Un diccionario de subcategorización
  - La raíz contiene los parámetros de ejecución
  - Entradas verbales
    - Entradas de característica estudiada
      - Patrón, cantidad de ocurrencia y umbral del test pasado
- Un diccionario de ID's
  - Por cada entrada verbal
    - Entrada de la característica estudiada
      - Patron
        - Lista de ID's encontrados en el corpus con ese patrón
- Un archivo con información de la ejecución

Ejemplo de corpus

```
<corpus>
  <oracion ID='1'>
    <palabra lema='el' sint='ART'>La</palabra>
    <palabra lema='casa' sint='NC'>casa</palabra>
    <palabra lema='ser' sint='V'>es</palabra>
    <palabra lema='lindo' sint='ADJ'>linda</palabra>
  </oracion>
   <oracion ID='2'>
          ....
   </oracion>
 </corpus>
```

Ejemplo de ejecución

Se ejecuta por línea de comando, es necesario tener python y xml.parsers.expat (que viene junto con python)

```
>python IRASubcat.py corpus.xml morfosint=V
   oracion lexical

>python IRASubcat.py corpus.xml morfosint=V
   oracion lema
```

Ejemplo de ejecución(2)

“Caminar	es	bueno	para	la	salud”
V	V	ADJ	PREP	ART	NC
caminar	ser	bueno	para	el	salud

Armamos el XML

```
<corpus>
   <oracion>
      <palabra sint='V' lema='caminar'>Caminar</palabra>
      <palabra sint='V' lema='ser'>es</palabra>
      <palabra sint='ADJ' lema='bueno'>bueno</palabra>
      <palabra sint='PREP' lema='para'>para</palabra>
      <palabra sint='ART' lema='el'>la</palabra>
      <palabra sint='NC' lema='salud'>salud</palabra>
   </oracion>
</corpus>
```

Configuración para el ejemplo

- Considerar una lista verbal=NO
- Actualizar un diccionario existente=NO
- Longitud a cada lado del verbo=3
- Completar el patrón con una palabra comodín=NO
- Tener en cuenta el orden=SI
- Características a estudiar=sint, sint-lema
- Usar la forma de la palabra=NO

Configuración para el ejemplo (cont.)

- Poner el carácter '|' en la posición del verbo=SI
- Colapsar patrones=NO
- Máximo número de iteraciones para colapsar los patrones=Falso
- Mínima frecuencia absoluta de verbo=0
- Mínima frecuencia de coocurrencia entre verbo y patrón=0
- Usar test de hipótesis likelihood ratio=SI

Diccionario de salida

```
<dictionary execute="IRASubcat.py corpus_oracion.xml
   sint=V oracion lema config.txt" verb_list="False"
   dict_exist="False" scope="3" comp_w_word="False"
   order="True" tags="['sint', 'sint-lema']"
   lex_items="False" verbal_mark="True"
   verb_min_abs_freq="0" pattern_min_abs_freq="0"
   colapse_patterns="False" use_likelihood="True">
  <entry verb="caminar" count_oc_verb="1">
    <tag name="sint" different_patterns="1">
      <pattern id="|,V,ADJ,PREP" count_w_verb="1"
         total_count="1"
         rejected_patterns_freq_test="NO"
         rejected_likelihood_test="'no_paso'">
      </pattern>
    </tag>
```

Diccionario de salida(cont.)

```
  <tag name="sint-lema" different_patterns="1">
    <pattern id="|,V-ser,ADJ-bueno,PREP-para"
       count_w_verb="1" total_count="1"
       rejected_patterns_freq_test="NO"
       rejected_likelihood_test="'no_paso'">
    </pattern>
  </tag>
</entry>
<entry verb="ser" count_oc_verb="1">
  <tag name="sint" different_patterns="1">
    <pattern id="V,|,ADJ,PREP,ART"
       count_w_verb="1" total_count="1"
       rejected_patterns_freq_test="NO"
       rejected_likelihood_test="'no_paso'">
    </pattern>
  </tag>
```

Diccionario de salida(cont.)

```
  <tag name="sint-lema" different_patterns="1">
    <pattern id="V-caminar,|,ADJ-bueno,PREP-
      para,ART-el" count_w_verb="1"
      total_count="1"
      rejected_patterns_freq_test="NO"
      rejected_likelihood_test="'no_paso'">
    </pattern>
  </tag>
 </entry>
</dictionary>
```

Detección de constituyentes opcionales

Colapsar patrones que comparten el mismo núcleo de constituyentes

Argumento verbal: constituyente requerido por un verbo para formar una oración gramatical

Adjunto: constituyente opcional

Ejemplo de colapsado

Consideremos las siguientes oraciones:

- La casa tiene un lindo patio con pileta
- La casa tiene un patio chico
- La casa tiene patio
- La casa tiene un gran patio con vista al mar
- Los departamentos tienen patios internos
- Las casas tienen grandes patios para descanzar

Test de hipótesis likelihood ratio

log-likelihood = 2[logL(p1, k1, n1) + logL(p2, k2, n2) - logL(p, k1, n1) + logL(p, k2, n2)]

donde:
n1 = cantidad de ocurrencias del verbo considerado
n2 = cantidad de ocurrencias de otros verbos
k1 = cantidad de ocurrencias del marco con el verbo considerado
k2 = cantidad de ocurrencias del marco con otro verbo

Test de frecuencia

- Umbral de frecuencia absoluta para verbos: Sirve para filtrar verbos con pocos ejemplos en el corpus
- Umbral de frecuencia relativa de co-ocurrencia de un patrón con un verbo: Sirve para descartar co-ocurrencias de patrones con verbos con pocos ejemplos en el corpus

Aplicación a un corpus del Español

- Corpus periodístico anotado a mano a varios niveles (morfológico, sintáctico, semántico)
- Anotación a nivel de constituyente y verbal
- Experimentamos con la función sintáctica
- Estudiamos 20 sentidos verbales:

  necesitar_1, lograr_1, escribir_1, tardar_1, anunciar_1, desear_1, afirmar_2, casar_1, merecer_1, negociar_1, votar_1, descartar_1, detectar_1, intentar_1, llenar_3, decidir_1, financiar_1, controlar_1, efectuar_1 y conseguir_1

Evaluando resultados

filtro aplicado	Precisión	Cobertura	medida-F
frecuencia	.79	.70	**.74**
likelihood ratio 90 %	.42	.46	.39
likelihood ratio 95 %	.38	.42	.32
likelihood ratio 99 %	.31	.36	.22
likelihood ratio 99.5 %	.25	.28	.14

Aplicación a un corpus del Ruso

Corpus plano, textos provenientes de periódicos rusos. 1.000.000 de palabras (500 MB)

Preprocesamiento con el TreeTagger
Se estudiaron los 10 patrones más frecuentes de: кушать, дать y спать

Resultados del Ruso

Para el verbo **кушать** se encontró que algunos patrones contenían los constituyentes esperados Nn Na, que son en ruso un Nominativo y un acusativo (Sujeto, Objeto directo)-Transitivo

Resultados del Ruso (cont.)

Verbo **дать**: como patrón de mayor frecuencia se obtuvo [Nn Na Nd] que corresponde a Sujeto Obj. Directo Obj. Indirecto (verbo transitivo c/Obj. Indirecto)

Verbo **спать**: en el 60% de los verbos aparece Nn (Sujeto) y en el 40% aparece R (constituyentes de sintagma adverbial) En español por lo general ocurre con circunstanciales y los sintagmas adverbiales son circunstanciales - intransitivo

Conclusiones

- Herramienta
  - Gratis de código abierto y licencia GPL
  - Independiente de lengua
  - Multiplataforma
  - Flexible
  - Parametrizable
  - Sin restricciones de tipo ni cantidad de marcos

Otros trabajos y trabajo en curso

- Traducción automática – Apertium, Cunei, Moses.
- Resolución de correferencias de pronombres personales
- Generación de Expresiones Referenciales (usando modelos lógicos)

Gracias!