Procesamiento del lenguaje natural
Técnicas lingüísticas formales

Índice

- Introducción
- Comprensión del lenguaje natural
  - Análisis morfo-léxico
  - Análisis sintáctico
    - DCGs
    - Gramáticas probabilísticas
  - Análisis semántico
    - Encaje de patrones (pattern-matching)
    - Gramáticas semánticas
    - Formas lógicas
    - Gramáticas de dependencias
  - Análisis pragmático
- Generación de lenguaje natural

INTRODUCCIÓN

Introducción

- El lenguaje natural es una forma de comunicación *imprecisa* y *ambigua* que se apoya en el conocimiento *compartido* por los que se comunican
  - Permite expresar una misma idea de muchas formas
  - El lenguaje natural está en continua expansión y cambio
- Procesamiento del lenguaje natural
  - Comprensión
  - Generación
  - Clasificación de texto
- Aplicaciones
  - Correctores gramaticales
  - Traducción automática
  - Búsqueda y filtrado de información
  - Sistemas de diálogo

Introducción

- Gran dificultad debido a que el lenguaje es algo vivo: en continua expansión, que va modificándose...
  - El lenguaje se modifica tanto en vocabulario como en sintaxis
  - Existencia de jergas locales, profesionales, por franjas de edad...

- Ambigüedad
  - Fonética: ¿Me diste la caja? ¿Mediste la caja?
    - Homofonía: Dos textos suenan iguales o prácticamente iguales
  - Léxica: *Entró en el banco. Se sentó en el banco.*
    - Polisemia: ambigüedad de las palabras
  - Sintáctica: *Juan vio a María con unos prismáticos.*
    - A veces es imposible de solucionar
  - Semántica: *Los niños compraron el libro de Peter Pan.*
  - Referencial: *El jamón está en el armario. Sácalo. Ciérralo.*

- Requiere mucho conocimiento: objetivos del hablante, hipótesis, contexto... No es mera transmisión de palabras...

Niveles de análisis

- Fonología
  - Combinación de sonidos (fonemas) que producen palabras
- Morfología
  - Estructura de las palabras (morfemas): género, número...
- Sintaxis
  - Combinación de palabras que producen frases
- Semántica
  - Significado de palabras + estructura de frase: significado de frase (normalmente, independientemente del contexto)
- Pragmática
  - Significado de frase + contexto: significado más profundo

COMPRENSIÓN DEL LENGUAJE NATURAL

Proceso de comprensión

- Comprensión: proceso de correspondencia de una forma de entrada a otra representación más útil para una cierta tarea
- Suele requerir 4 fases (o 5 si el lenguaje es hablado)
  - Estas fases suelen tener límites difusos
  - **No todos los sistemas trabajan con todas fases necesariamente**
  - Análisis fonológico
    - Este nivel se aplica sólo si el origen del texto es hablado.
    - Interpretación de los sonidos del habla dentro de las palabras y entre ellas.
    - Los sonidos a menudo informan del significado o intención de una palabra o una frase
    - Transforma los sonidos del habla en las palabras que mejor encajan
  - Análisis morfo-léxico
    - El texto se separa en unidades léxicas con significado (lexemas)
      - Pueden ser fragmentos de una palabra, una o varias palabras
    - Se obtienen los lexemas y morfemas gramaticales
    - Se les asignan una o varias categorías sintácticas (si hay ambigüedad)

Proceso de comprensión

- Análisis sintáctico
  - Se comprueba la corrección de una frase con respecto a las reglas del lenguaje (gramática)
  - Se convierte la secuencia (lineal) de unidades léxicas en estructuras sintácticas (no lineales)
- Análisis semántico
  - Se asigna significado a una frase
  - Correspondencia entre estructuras sintácticas y objetos del dominio representados en algún lenguaje de representación del conocimiento
- Análisis pragmático
  - Hasta esta fase el análisis se hace a nivel de frase. Aquí, pasan a considerarse párrafos enteros o incluso textos de forma global
  - La mayoría de los sistemas reales no la incluyen
    - Líneas de investigación abiertas
  - Se resuelven las referencias relativas al contexto (ambigüedades referenciales, como por ej. referentes de un pronombre) y se asigna significado global a un conjunto de frases

COMPRENSIÓN: ANÁLISIS MORFO-LÉXICO

Análisis morfo-léxico

- Se basa en
  - Diccionarios (lexicones)
  - Reglas morfológicas (cómo se forman las derivaciones de palabras)
- Ambas cosas son interdependientes
  - Si en el diccionario sólo guardamos lexemas, necesitaremos muchas reglas morfológicas
  - Si guardáramos todas las formas de las palabras en el diccionario, no necesitaríamos reglas morfológicas
  - Hay que llegar a un compromiso entre ambos extremos
- Dificultades
  - Polisemia: palabra con varios significados
    - Ej.: banco (dinero, sentarse, río, peces)
  - Homonimia: palabras distintas con la misma grafía
    - Ej.: divorciado (nombre, adjetivo y verbo)

Análisis morfo-léxico

- Suele incluir 4 etapas:
  - Descomposición del texto en palabras y signos de puntuación
  - Análisis morfológico: descomponer una palabra en su raíz (lexema), prefijos y sufijos, siguiendo 3 tipos de reglas:
    - Flexión: plural, femenino, conjugaciones de los verbos, declinaciones según el caso *(yo, me)*
    - Derivación: a partir de otra de diferente categoría *(adjetivo → adverbio)*
    - Composición *(limpia/parabrisas)*
  - Búsqueda en diccionario: anterior o posterior al análisis morfológico
  - Tratamiento de errores
    - Si no se encuentra en el diccionario, a veces se puede establecer su categoría por la terminación *("mente" → adverbio)*
    - Si empieza por mayúscula se puede considerar nombre propio
    - Puede haber formatos especiales (fechas, DNI,...)
    - *Verificadores ortográficos*. Palabras que se diferencian en una letra o con letras intercambiadas

Diccionarios

- En los diccionarios se guardan distintos tipos de información:
  - Categoría sintáctica
    - Categorías cerradas (preposiciones, conjunciones, etc.)
    - Categorías abiertas (nombre, adjetivo, verbo)
    - Si una palabra tiene varias categorías, tiene que aparecer para cada una de ellas
  - Concordancia
    - Género, número, persona, caso
  - Preposiciones que admite un verbo, tipos de complementos, etc.
  - Información morfológica (patrón de formación de la palabra)
  - Información semántica
    - Concepto correspondiente, palabras sinónimas
    - Comprensión:
      - Identificar con qué concepto se corresponde la palabra
      - Varias acepciones → varios conceptos

Diccionarios

- Los diccionarios se suelen organizar utilizando relaciones de herencia múltiple, tanto de tipo gramatical como conceptual
- La construcción se realiza de forma manual o semiautomática a partir de una ontología de conceptos bajo la cual se van colocando las palabras
  - Son muy costosos de generar y mantener
  - Requieren la colaboración de muchos expertos
  - Son un recurso muy útil en la comprensión del lenguaje

WORDNET (Princeton University)

- Base de datos léxica del inglés organizada semánticamente
  - http://wordnet.princeton.edu/
- La 1ª versión (1993) contenía 95.600 palabras o grupos de palabras organizadas en 70.100 significados diferentes
  - (la actual: 155.287 y 117.659)
- Relación entre palabras por distintas relaciones semánticas:
  - sinonimia, antonimia
  - es-un: hiponimia, hiperonimia *(inversa de es-un)*
  - parte-de: meronimia, holonimia *(inversa de parte-de)*
- Organización de palabras en una jerarquía de unos 12 niveles que permite la herencia (basada en ideas de redes semánticas)
  - Hay 4 categorías sintácticas: nombre, verbo, adjetivo **y** adverbio
  - Hay palabras que están repetidas en más de una categoría sintáctica (homonimia)
  - Una palabra puede tener varios significados (polisemia) y para cada uno de ellos se da una lista de sinónimos (synset: los conceptos)

WORDNET (Princeton University)

- Elementos principales de WordNet
  - Conceptos = conjuntos de palabras sinónimas (*synset*)

    car $\in$ {car, auto, automobile, machine, motorcar}
  - Relaciones importantes
    - Sinonimia-antonimia

      rapidly/speedly $\leftrightarrow$ slowly
    - Hiponimia-hipernimia (sub/super-concepto)

      {car, auto, automobile, machine, motorcar}

      isa {motor vehicle, automotive vehicle}

      isa {vehicle}
    - Meronimia (parte/miembro)

      {car, auto, automobile, machine, motorcar}

      hasa {automobile engine}

      hasa {bumper}

WORDNET (Princeton University)

- WordNet (investigation)
  - Sense 1: probe, probing, investigation -- (an inquiry into unfamiliar or questionable activities; "there was a congressional probe into the scandal")
    - inquiry, enquiry, research -- (a search for knowledge; "their pottery deserves more study than it has received")
  - Sense 2: investigation, investigating -- (the work of inquiring into something thoroughly and systematically)
    - work -- (activity directed toward making or doing something; "she checked several points needing further work")
  - http://wordnetweb.princeton.edu/perl/webwn

- Se han desarrollado versiones de WordNet en muchos otros idiomas
  - http://globalwordnet.org/wordnets-in-the-world/

COMPRENSIÓN: ANÁLISIS SINTÁCTICO

Análisis sintáctico

- "Deslinearización" de una frase, determinando las funciones que realizan las palabras que la componen, para obtener así una estructura jerárquica que permita asignarle significado
- El significado de una frase depende en gran medida del orden de sus palabras
- Cada lenguaje natural tiene sus propias reglas en cuanto a aceptar las posibles combinaciones de palabras como frases válidas
  - Estas reglas se suelen representar en forma de gramáticas
- La mayoría de los sistemas realizan análisis sintáctico usando
  - Una gramática: representación declarativa de la sintaxis del lenguaje
  - Un analizador sintáctico *(parser)*: programa que compara las reglas de la gramática con las frases a analizar para producir las estructuras correspondientes

Gramáticas

- Una gramática G se define como una cuádrupla (VN, VT, P, S) donde
  - VN: vocabulario no terminal
  - VT: vocabulario terminal (disjunto del anterior)
  - P: conjunto finito de producciones (reglas de reescritura)
  - S: símbolo inicial perteneciente a VN
  - Cada producción es de la forma $X \rightarrow Y$, siendo
    - X e Y cadenas de símbolos de $V \equiv VN \cup VT$
    - X cadena no vacía
- Una gramática describe las estructuras válidas del lenguaje que denota
  - Se usan aquí para el análisis sintáctico de las frases
  - El resultado puede ser uno o varios árboles de análisis sintáctico

Tipos de gramáticas

- Jerarquía de Chomsky (1957):
  - Gramáticas tipo 0 (Recursivamente enumerables)
    - Sin restricciones en cuanto a la forma de las producciones
    - Equivalentes en potencia expresiva a las máquinas de Turing
  - Gramáticas tipo 1 (Dependientes del contexto)
    - $\alpha A \beta \rightarrow \alpha \gamma \beta$ siendo $A \in VN$, $\alpha, \beta \in V^*$, $\gamma \in V^+$
  - Gramáticas tipo 2 (Independientes del contexto)
    - $A \rightarrow \gamma$ siendo $A \in VN$, $\gamma \in V^*$
    - Equivalentes en potencia a los autómatas de pila
  - Gramáticas tipo 3 (Regulares)
    - $A \rightarrow a$ o $A \rightarrow a\ B$ siendo $A, B \in VN$, $a \in VT$
    - Formato equivalente: $A \rightarrow w$ o $A \rightarrow w\ B$ siendo $w \in VT^*$ (lineales por la derecha).
    - Equivalentes en potencia a los autómatas finitos

Gramáticas para lenguajes naturales

- Las gramáticas independientes del contexto (tipo 2) son muy utilizadas como gramáticas para LPs y lenguajes naturales
  - Hay construcciones de algunos lenguajes naturales que no son independientes del contexto (p.ej., los fenómenos de concordancia)
  - La mayor parte de su estructura sí lo es
- Las regulares (tipo 3) se ajustan peor, incluso a los LPs
  - Imposibilidad de capturar anidamientos, por ejemplo
- Existen muchos mecanismos eficientes de *parsing* para este tipo de gramáticas (tipo 2 y 3)
- Lo más habitual en el procesamiento del lenguaje natural es utilizar gramáticas independientes del contexto con algún tipo de extensión
  - Estas características adicionales las dotan de mayor capacidad expresiva y cambian su lugar en la jerarquía

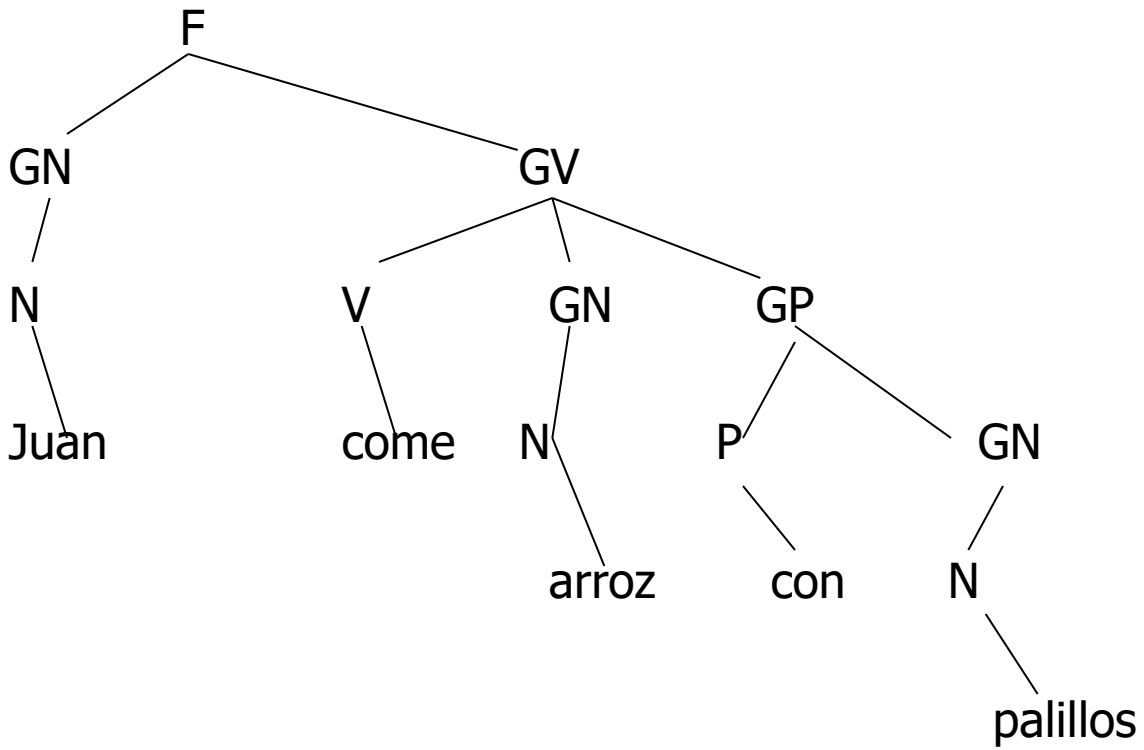
Tipos de análisis sintáctico

- Método descendente (razonamiento hacia delante)
  - Empieza a partir del símbolo inicial y va aplicando producciones hasta que se ha desarrollado el árbol de derivación y los símbolos terminales que etiquetan a las hojas se corresponden con la frase
- Método ascendente (razonamiento hacia atrás)
  - El analizador empieza por los símbolos terminales de la frase, viendo a qué categoría corresponden y, aplicando las reglas al revés, intenta alcanzar el símbolo inicial
- La elección suele depender
  - En primer lugar, del factor de ramificación
  - También de si hay heurísticas aplicables para desechar caminos
- Método híbrido: ascendente con filtro descendente
- La salida puede no ser una única estructura
  (si hay ambigüedad sintáctica y se quieren varias respuestas)

Ambigüedad sintáctica

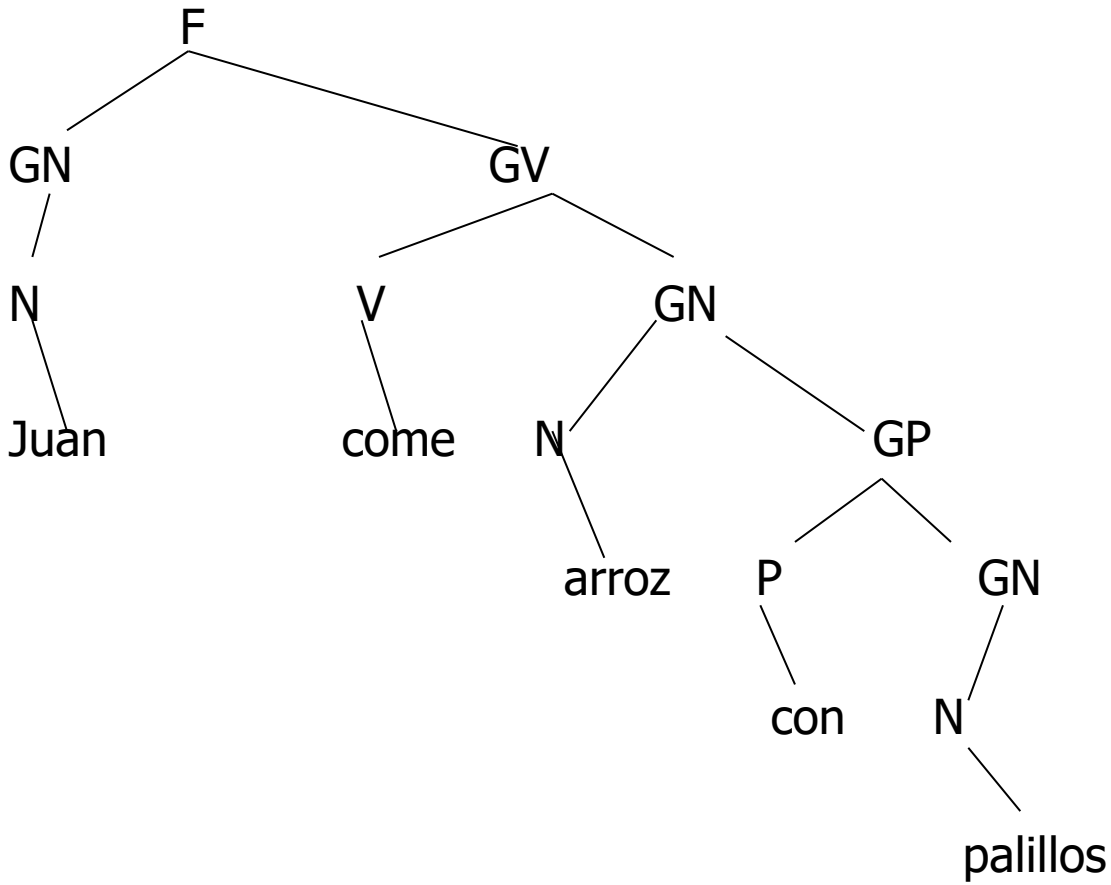
- Ejemplo
  - "Como todos los días..."
  - No se ha analizado toda la frase: ¿conjunción o verbo?
  - Podría aclararse al seguir leyendo
    - "Como todos los días llegas tarde" / "Como todos los días ensalada"
  - Hay varias alternativas para lidiar con esto en la implementación de los analizadores sintácticos
- Pero hay frases "genuinamente" ambiguas
  - "Estaba en casa cuando llamó"
  - ¿1ª o 3ª persona?
  - O se siguen todos los caminos
    - Costoso por mantener estructuras intermedias que se desecharán
  - O se contempla el backtracking
    - Costoso también por cómputos duplicados
  - Lo más habitual: una sola interpretación plausible en esta fase

Ambigüedad sintáctica



- Este análisis es correcto desde el punto de vista sintáctico. Además es válido semánticamente porque los palillos son un instrumento y no algo comestible

Ambigüedad sintáctica



- Este análisis es correcto desde el punto de vista sintáctico, pero no desde el punto de vista semántico. Lo sería para una frase como "Juan come arroz con pimientos"

Gramáticas de unificación

- Las gramáticas de unificación o gramáticas lógicas utilizan el mecanismo de unificación como base para el proceso de análisis
- Las gramáticas de unificación se pueden utilizar, además de en el análisis sintáctico, en otros análisis como el morfológico y el semántico
- Las gramáticas de cláusulas definidas (DCGs) son uno de los tipos de gramáticas de unificación más utilizados
  - Forman parte del Prolog estándar
  - Se concibieron como una generalización ejecutable de las GICs

DCGs (Gramáticas de cláusulas definidas)

- Las DCGs son un formalismo de representación de gramáticas de unificación que permite representar directamente gramáticas independientes del contexto (tipo 2)
  - Se usa --> como operador para representar la → de las reglas
  - Los símbolos no terminales deben ser átomos
  - Los símbolos terminales deben ser átomos también
    - Para distinguirlos de los no terminales, una secuencia de terminales se escribe como una lista (la cadena vacía es [ ])
  - Se separan los símbolos mediante comas
  - Las reglas terminan en punto

- Las DCGs incluyen una serie de extensiones que permiten representar características dependientes del contexto y que les confieren una potencia expresiva equivalente a las máquinas de Turing

DCGs (Gramáticas de cláusulas definidas)

- Ejemplo:

```
frase --> grupo_nominal, grupo_verbal.
grupo_nominal --> nombre.
grupo_verbal --> verbo, grupo_nominal.
grupo_verbal --> verbo.
nombre --> [manzanas].
nombre --> [juan].
verbo --> [come].
```

  - Corresponde a una GIC G = (VN, VT, P, S) donde
    - VN = {frase, grupo_nominal, grupo_verbal, nombre, verbo}
    - VT = {manzanas, juan, come}
    - S = frase
    - P contiene las producciones anteriores

Ejemplo de uso

```
frase --> grupo_nominal, grupo_verbal.
grupo_nominal --> nombre.
grupo_verbal --> verbo, grupo_nominal.
grupo_verbal --> verbo.
nombre --> [manzanas].
nombre --> [juan].
verbo --> [come].
```

- Para analizar una frase completa, escribiríamos

  ```
  ?- frase([juan, come, manzanas], []).
  ```

- Es un reconocedor y no un analizador (porque no devuelve la estructura sintáctica)
- En esta DCG no está representada la concordancia
- También reconoce la frase “manzanas come Juan”

Traducción de DCGs a predicados Prolog

- Cada símbolo no terminal se convierte en un predicado binario
  - El primer parámetro representa la cadena antes de reconocer el símbolo
  - El segundo parámetro representa lo que queda de cadena tras reconocer el símbolo
  - Se implementa usando un mecanismo eficiente llamado listas diferencia
- Las cadenas de símbolos terminales se representan como listas

```
% frase --> grupo_nominal, grupo_verbal.
frase(A, B) :- grupo_nominal(A, C), grupo_verbal(C, B).
…
% nombre --> [manzanas].
nombre([manzanas | A], A).
```

- Por eso para analizar una frase se escribe el predicado así:

```
?- frase([juan, come, manzanas], []).
```

DCGs

- El uso de DCGs no requiere conocer su traducción a Prolog
  - Siempre podemos comprobar el proceso automático de traducción a cláusulas Prolog usando el predicado **listing**
- Para comprobar si una frase es correcta añadimos como segundo parámetro la lista vacía:

```
?- frase([juan, come, manzanas], []).
```

- También podemos generar todas las frases "correctas" (*derivables)*:

```
?- frase(X, []).
X= [manzanas, come, manzanas] ;
X= [manzanas, come, juan] ; X= [manzanas, come] ;
X= [juan, come, manzanas] ; X= [juan, come, juan] ;
X= [juan, come] ;
No
```

DCGs: extensiones con respecto a las GICs

- Las extensiones a las GICs que incorporan las DCGs suponen añadirles una serie de características que pueden resumirse en
  - Posibilidad de efectuar llamadas a predicados Prolog (escritos entre llaves { })
  - Posibilidad de añadir parámetros a los símbolos no terminales
  - Potencia de la unificación

- Esto aumenta su utilidad y potencia expresiva
  - Permiten representar restricciones de concordancia, obtener el árbol de análisis y facilitar otras tareas relacionadas con el análisis

- Los símbolos no terminales, que pueden tener ahora aridad *n*, se convierten en predicados *n*+*2*-arios
    - Los 2 parámetros **últimos** son la cadena a procesar antes y después de reconocer el símbolo.

Parámetros en símbolos no terminales: concordancia

- Ejemplo de uso para garantizar la concordancia entre sujeto y verbo

```
frase --> grupo_nominal(Numero),  grupo_verbal(Numero).
grupo_nominal(Numero) --> nombre(Numero).
grupo_verbal(Numero) --> verbo(Numero),
                         grupo_nominal(Numero1).
%Complemento Directo: no concuerda con el verbo
grupo_verbal(Numero) --> verbo(Numero).
nombre(plural) --> [manzanas].
nombre(singular) --> [juan].
verbo(singular) --> [come].
```

- Mismo uso:

```
?- frase([juan, come, manzanas], []).
```

Parámetros en símbolos no terminales: concordancia

```
frase --> grupo_nominal(Num), grupo_verbal(Num).
grupo_nominal(Numero) --> nombre(Numero).
grupo_verbal(Numero) --> verbo(Numero),  grupo_nominal(Numero1). %CD
grupo_verbal(Numero) --> verbo(Numero).
nombre(plural) --> [manzanas].
nombre(singular) --> [juan].
verbo(singular) --> [come].
```

- Uso para generación

```
?- frase(Xs, []).
Xs = [juan, come, manzanas] ;
Xs = [juan, come, juan] ;
Xs = [juan, come] ;
No
```

Parámetros en símbolos no terminales: obtener el árbol de análisis

- Ejemplo de uso para obtención del árbol de análisis

```
frase(f(GN, GV)) --> grupo_nominal(GN), grupo_verbal(GV).
grupo_nominal(gn(N)) --> nombre(N).
grupo_verbal(gv(V, GN)) --> verbo(V), grupo_nominal(GN).
grupo_verbal(gv(V)) --> verbo(V).
nombre(n(manzanas)) --> [manzanas].
nombre(n(juan)) --> [juan].
verbo(v(come)) --> [come].
```

- Uso para obtener el árbol de derivación de una frase

```
?-frase(Arbol, [juan, come, manzanas], []).
Arbol= f(gn(n(juan)),gv(v(come),gn(n(manzanas))))
```

Parámetros en símbolos no terminales: obtener el árbol de análisis

- Generación de frases y árboles

```
?- frase(Arbol, Xs, []).
Arbol = f(gn(n(manzanas)),gv(v(come),gn(n(manzanas))))
Xs = [manzanas, come, manzanas] ;
Arbol = f(gn(n(manzanas)), gv(v(come), gn(n(juan))))
Xs = [manzanas, come, juan] ;
Arbol = f(gn(n(manzanas)), gv(v(come)))
Xs = [manzanas, come] ;
Arbol = f(gn(n(juan)), gv(v(come), gn(n(manzanas))))
Xs = [juan, come, manzanas] ;
Arbol = f(gn(n(juan)), gv(v(come), gn(n(juan))))
Xs = [juan, come, juan] ;
Arbol = f(gn(n(juan)), gv(v(come)))
Xs = [juan, come] ;
No
```

Llamadas a predicados

- Otra extensión consiste en incorporar requisitos de satisfacción de predicados Prolog en la parte derecha de las reglas
- Estos predicados actúan como “llamadas a procedimientos externos a la gramática”
- Deben distinguirse sintácticamente para evitar que sean considerados como categorías sintácticas por el preprocesador de DCGs
  - Si no, les añadiría los dos argumentos habituales, y probablemente resultaría en un error de ejecución
  - Por ello las llamadas a predicados Prolog se colocan entre llaves en las reglas
  - Su traducción a Prolog es directa: se mantienen tal cual (eliminando las llaves)

Llamadas a predicados: uso de diccionarios

- Ejemplo de uso en construcción de diccionarios:
  - En vez de

```
nombre(n(manzanas)) --> [manzanas].
nombre(n(juan)) --> [juan].
```

  - Conviene poner

```
nombre(n(P)) --> [P], {es_nombre(P)}.
es_nombre(manzanas).
es_nombre(juan).
```

- De esta forma, la implementación del diccionario se hace en forma de hechos y no mediante reglas que harían crecer innecesariamente el tamaño de la gramática
  - No es razonable tener una regla por cada palabra

Llamadas a predicados: uso de diccionarios

- Ejemplo de uso para comprobación de preposiciones permitidas por verbos *(sub-categorización de verbos)*

```
frase --> nombre_propio, verbo(X), complemento(X).
complemento([]) --> [].
complemento([X]) --> preposición(X), nombre_propio.
verbo(X) --> [V], {es_verbo(V, X)}.
nombre_propio --> [N], {es_nombre_p(N)}.
preposición(P) --> [P], {es_prepo(P)}.
```

```
es_verbo(piensa, [en]).      es_nombre_p(juan).   es_prepo(en).
es_verbo(ríe, []).           es_nombre_p(maría).  es_prepo(con).
es_verbo(habla, [con]).      es_nombre_p(ana).
```

- ¿Qué tipo de frases permitiríamos reconocer con esta gramática?

Llamadas a predicados: uso de diccionarios

- Generación de las frases que reconocería la DCG anterior con

```
?- frase(Xs, []).
```

- Las siguientes combinaciones se reconocerían como correctas
  - Juan/María/Ana piensa en Juan/María/Ana
  - Juan/María/Ana ríe
  - Juan/María/Ana habla con Juan/María/Ana

Gramáticas probabilísticas

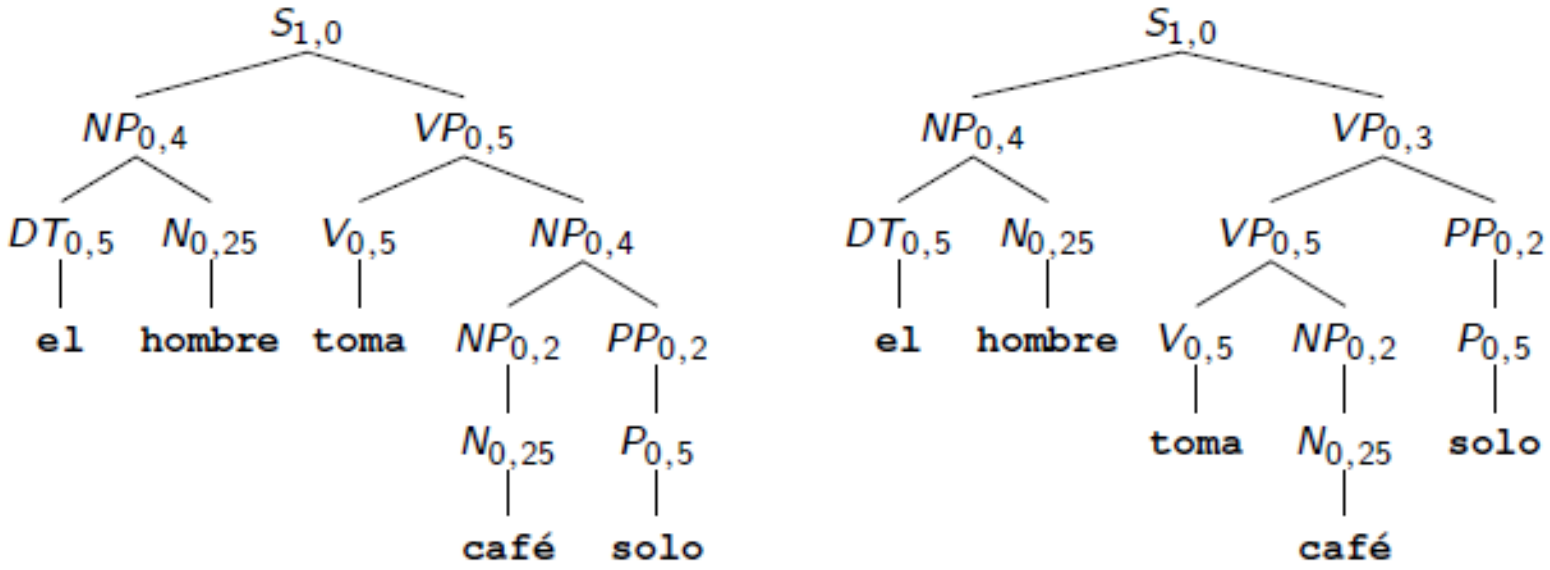
- Permiten desambiguar asociando una probabilidad a cada posible árbol de derivación de una frase

- Cada regla de la gramática tiene asociada una cierta probabilidad que refleja su frecuencia relativa de uso
  - Las probabilidades se extraen a partir de un corpus de oraciones
  - La suma de las probabilidades de las reglas con el mismo antecedente será 1
- Ejemplo:
  - O -> S, P (0.8) S -> Det, N (0.40) S -> Det, N, Adj (0.25) …

- P(frase) = π P(reglas aplicadas en el análisis)
  - P(el perro ladra) = P(O -> S, P) * P(S -> Det, N) * P(P->V) * P(Det->el) * P(N->perro) * P(V->ladra)

Gramáticas probabilísticas

- Si hay dos posibles análisis (ambigüedad) => se calcula la probabilidad de cada árbol para determinar el más probable



  - Probabilidad del primer ejemplo: 0,000025
  - Probabilidad del segundo ejemplo: 0,0000187

COMPRENSIÓN: ANÁLISIS SEMÁNTICO

Análisis semántico

- El árbol de análisis sintáctico de una frase es tan sólo el primer paso hacia la comprensión de la frase
- Hay que producir una representación de su significado
  - Comprensión del LN: proceso de correspondencia de una forma de entrada a otra representación de salida útil para una cierta tarea
  - No existe “El Lenguaje” en el que representar todos los significados
    - Los distintos sistemas de representación de conocimiento podrían ser potenciales candidatos (lógica de predicados, sistemas de marcos…)
    - Y aún nos faltaría definir el vocabulario que se utilizará (los predicados concretos, los marcos o lo que corresponda...)
  - Llamaremos a este lenguaje final, sea el que sea, lenguaje objeto
  - Su elección depende de lo que haya que hacer con los significados una vez construidos

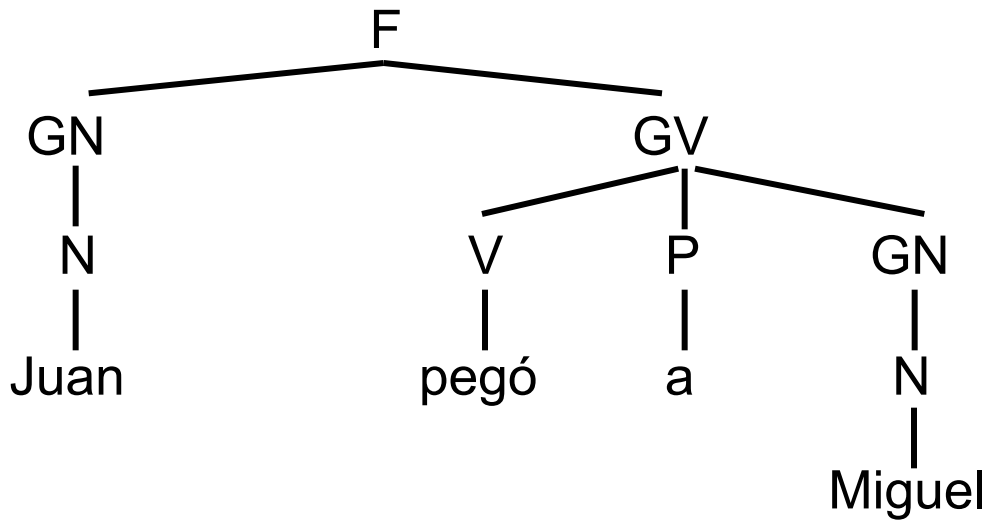
Análisis semántico

- El objetivo de interpretación semántica que se plantea depende totalmente de la aplicación a desarrollar

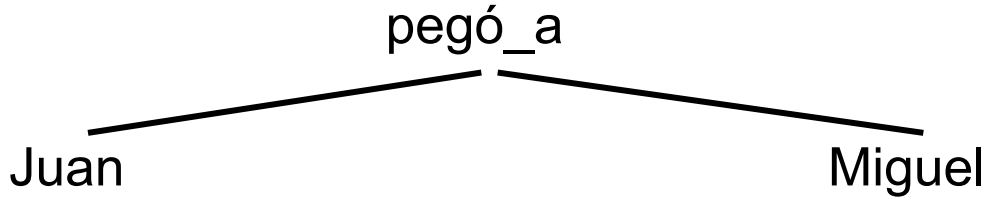
Aplicación	Salida
Interfaz en LN a Base de Datos	Lenguaje de consulta BD
Interfaz Sistema Operativo	Órdenes del SO
Sistema de diálogo	Respuesta adecuada en LN
Analizador de noticias	Relleno de una plantilla
Sintetizador de textos	Texto en LN
Traductor Automático	Texto en LN
Sistema de comprensión de texto	Lenguaje de representación de conocimiento

Análisis sintáctico vs. análisis semántico

- Ejemplo: Juan pegó a Miguel
  - Ejemplo de árbol de análisis sintáctico



  - Ejemplo de árbol de análisis semántico
    - El significado del verbo es la parte central de la semántica



Análisis semántico

- Tipos de aplicaciones
  - Más simples: salida producida directamente a partir de la entrada
  - Más complejas: se usa representación intermedia del significado
- Análisis semántico
  - Procesamiento léxico: buscar las palabras en un diccionario y desambiguar (en lo posible) teniendo en cuenta el contexto local
    - Las conexiones estructurales entre la estructura sintáctica y la semántica permiten seleccionar entre los distintos posibles resultados del análisis sintáctico sólo aquellos cuyo significado tenga sentido
  - Procesamiento a nivel de oración: representación semántica
- Algunos de los métodos utilizados
  - Encaje de patrones *(pattern-matching)*
  - Gramáticas semánticas
  - Formas lógicas
  - Gramáticas de dependencias

Encaje de patrones

- Es un mecanismo muy primitivo utilizado en sistemas como ELIZA (simulaba a un psicólogo)
- No se puede considerar una verdadera interpretación semántica, ya que sólo daba respuestas aparentemente adecuadas en función de los patrones que había reconocido
  - No hay comprensión

Patrón	Respuesta
X always Y	**Can you think of a specific example?**
X you Y me	**Why do you think I Y you?**
I remember X	**Why do you remember X just now?**
My {family member} is Y	**Who else in your family is Y?**
Yes, he is	**You seem very positive**

Encaje de patrones

Patrón	Respuesta
X always Y	Can you think of a specific example?
X you Y me	Why do you think I Y you?

Paciente: People always stare at me

Eliza: Can you think of a specific example?

Paciente: For instance you keep staring at me

Eliza: Why do you think I keep staring at you?

- En inglés funciona mejor por el tipo de morfología del idioma

Gramáticas semánticas

- Incorporan conocimiento sintáctico, semántico y pragmático en un conjunto de reglas que constituye una gramática
  - El resultado del análisis es una descripción semántica de la frase
- Las categorías son semánticas en vez de sintácticas
  - La posición de la palabra en la frase se usa para dar significado
  - Suele haber una acción semántica asociada a cada regla
- Simplifican muchos aspectos (concordancias, ...)
- Se utilizan en interfaces a programas como bases de datos, sistemas expertos, sistemas operativos, etc.
  - Por ejemplo, el sistema LIFER para acceso a una base de datos sobre barcos
  - En la pregunta hay que identificar por qué atributo se está preguntando y por qué barco concreto

Gramáticas semánticas: Sistema LIFER

- Gramática semántica *(sin acciones semánticas asociadas)*:

  pregunta --> comienzo, [the], atributo, [of], barco.

  comienzo --> [what,is]; [tell,me]; [can,you,tell,me].

  atributo --> [length]; [beam]; [class].

  barco --> [the], nombrebarco; nombreclase, [class, ship].

  nombrebarco --> [kennedy]; [enterprise].

  nombreclase --> [kitty,hawk]; [lafayette].

  atributo y barco son lo que interesa *(no son categorías sintácticas)*

- Ejemplos de preguntas admitidas:

  What is the length of the kennedy

  Tell me the class of the enterprise

  What is the beam of lafayette class ship

Gramáticas semánticas: Sistema LIFER

- En Prolog:

  ?- pregunta([what, is, the, length, of, the, kennedy], []).

- La idea es identificar la característica y el barco en la pregunta, consultar una base de datos y mostrar la respuesta.

- Supongamos que la base de datos también está implementada en Prolog

  barco_atrib_val(kennedy, length, 1200).

  barco_atrib_val(kennedy, beam, 250).

  barco_atrib_val(kennedy, class, lafayette).

  barco_atrib_val(enterprise, length, 1100).

  ...

Gramáticas semánticas: Sistema LIFER

- La gramática quedaría *(con acciones semánticas):*

```
pregunta(Nombre, Atributo) -->
                    comienzo, [the], atributo(Atributo), [of], barco(Nombre).
comienzo --> [what,is]; [tell,me]; [can,you,tell,me].
atributo(Atributo) --> [Atributo], {es_atributo(Atributo)}.
barco(Nombre) --> [the], nombre_barco(Nombre);
                          nombre_clase(Nombre), [class,ship].
nombre_barco(Nombre) --> [Nombre], {es_nombre_barco(Nombre)}.
nombre_clase(Nombre) --> [Nombre], {es_nombre_clase(Nombre)}.
```

- Y el "diccionario" así:

```
es_atributo(Atributo):- barco_atrib_val(_, Atributo, _), !.
es_nombre_barco(Nombre):- barco_atrib_val(Nombre, _, _), !.
es_nombre_clase(Nombre):- barco_atrib_val(_, class, Nombre), !.
```

Gramáticas semánticas: Sistema LIFER

- Una vez analizada la pregunta hay que hacer la consulta a la B.D.
  barco_atrib_val(Nombre, Atributo, Val) y escribir **Val**

- Por ejemplo

```
procesa_pregunta(Xs) :-
        pregunta(Nombre, Atributo, Xs, []),
        barco_atrib_val(Nombre, Atributo, Val),
        write(El valor es ), write(Val).

?- procesa_pregunta([what,is,the,length,of,the,kennedy]).
El valor es 1200
```

Gramáticas semánticas: ventajas e inconvenientes

- Método eficiente para la interpretación del LN en dominios reducidos, eliminando mucha ambigüedad sintáctica

- Carecen de abstracciones sintácticas, por lo que muchas construcciones similares se repiten en distintas categorías semánticas. Debido a ello, crecen demasiado rápidamente
  - Al poder necesitarse muchas reglas, el proceso de análisis puede resultar muy costoso

- Evolucionan hacia la integración de abstracciones sintácticas y semánticas en la misma gramática
  - Se evitan las ambigüedades de análisis estrictamente sintácticas teniendo en cuenta la semántica

Gramáticas semánticas: ventajas e inconvenientes

- Tras muchos experimentos de uso de gramáticas semánticas en una gran variedad de dominios, las conclusiones fundamentales son:
  - Gran utilidad para producir interfaces restringidas para el LN muy rápidamente
  - Pero no ofrecen una solución global al problema de la comprensión del lenguaje natural
    - Imposibilidad de capturar generalizaciones lingüísticas importantes

Formas lógicas

- El objetivo es representar el significado de la oración utilizando lógica de primer orden para poder "razonar" sobre el conocimiento representado

- Ejemplos:
  - Juan es alto
    - alto(juan)
  - Pedro bebe agua
    - bebe(pedro, agua)
  - Todo hombre tiene alma
    - ∀x[hombre(x) -> tiene(x, alma)]
  - Algún hombre tiene dinero:
    - ∃x[hombre(x) -> tiene(x, dinero)]

Formas lógicas

- "Juan es alto"
  - "Juan" es el término constante juan
  - "es" es un nexo de unión entre sujeto y adjetivo
  - "alto" es el predicado unario *alto* que expresa una propiedad sobre alguien
    - Se representa normalmente como $\lambda x.alto(x)$

- Significado de la frase
  - Se obtiene aplicando el significado del sintagma verbal al significado del sintagma nominal
  - $(\lambda x.alto(x))(juan) = alto(juan)$

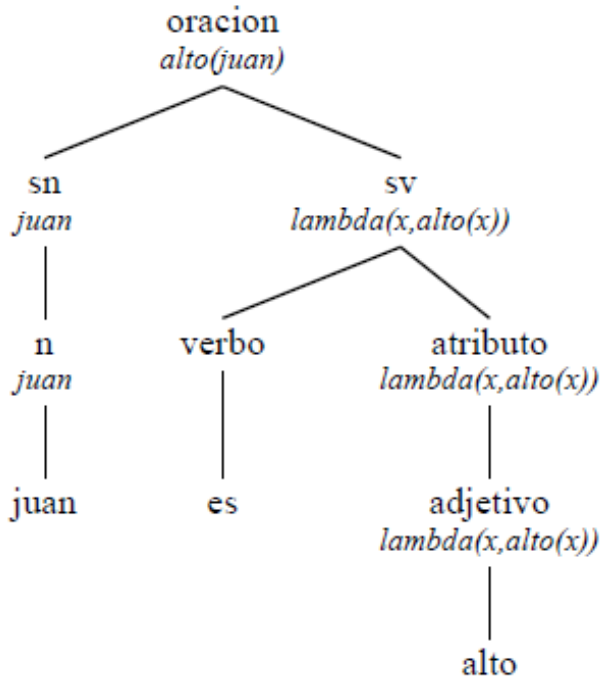
Formas lógicas

- La interpretación se basa en la Semantica Composicional
  - El significado de una expresión compleja depende de los significados de sus partes y de la reglas sintácticas mediante las que se combinan
  - La unidad básica es el sentido de una palabra, o en general de una unidad léxica
  - Esta hipótesis no siempre es cierta, pero facilita el análisis semántico

Formas lógicas

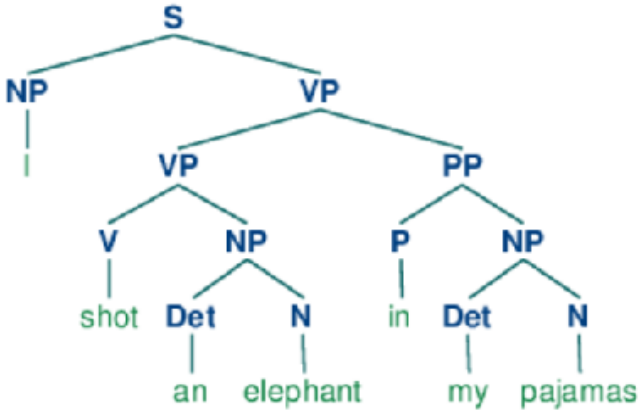
Procedimiento para el proceso composicional

- A cada nodo del árbol de análisis se le asigna una FL, que corresponde a la interpretación de la secuencia de palabras derivadas a partir de ese nodo.
- La FL de ese nodo se calcula a partir de las FL asociadas a los nodos hijos.
- La FL asociada al nodo raíz será la interpretación de la oración.

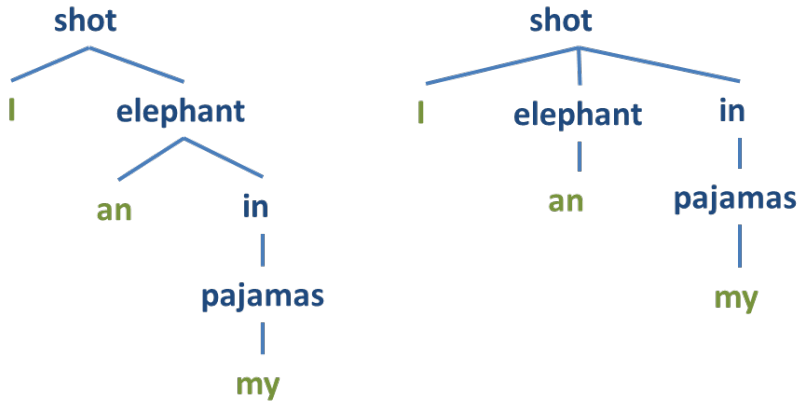


Gramáticas de dependencias

- Las gramáticas tradicionales se basan en los grupos sintácticos que constituyen las frases y sus relaciones (sujeto, predicado, complemento directo, etc.)
- Las gramáticas de dependencias son otro enfoque diferente y complementario que se basa en analizar las relaciones de dependencia entre pares de palabras.



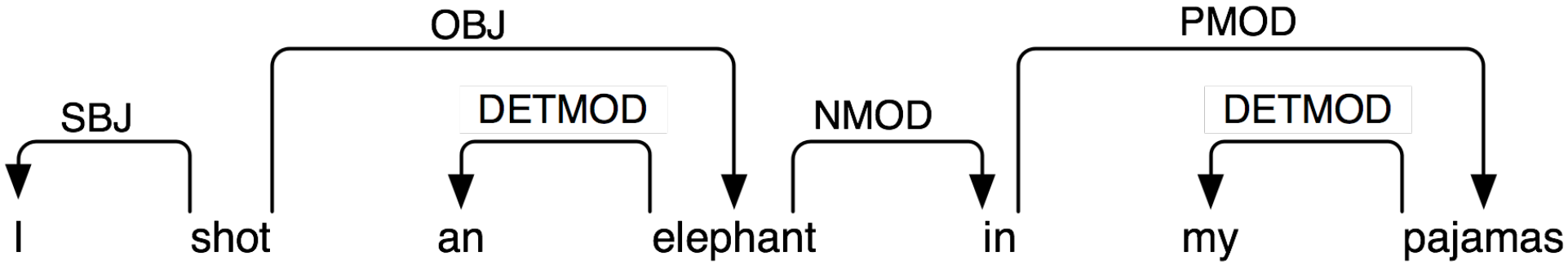
Gramática sintáctica



Gramática de dependencias

Gramáticas de dependencias

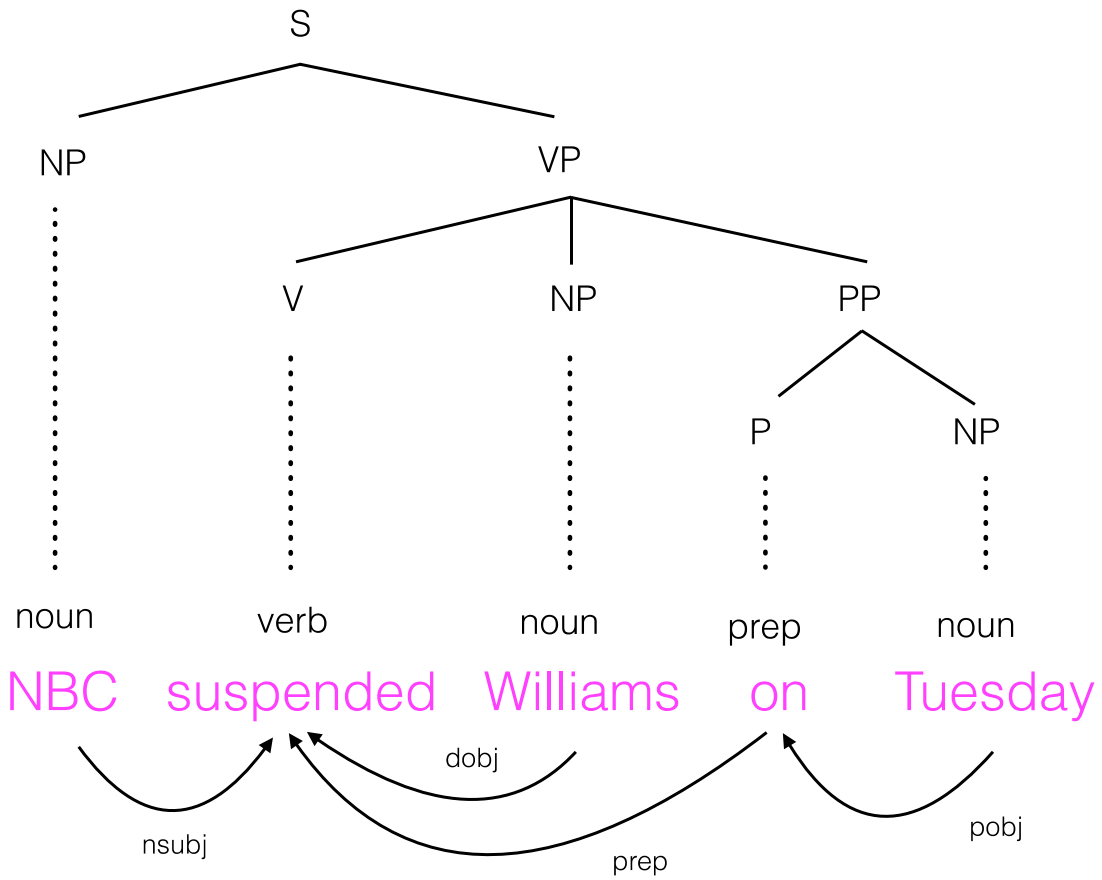
- No tienen símbolos de variable, las relaciones se establecen directamente entre palabras (no entre grupos de palabras)
- Relaciones binarias donde una palabra es la cabeza (*head*) y la otra la parte dependiente (*dependent*).
- El tipo de relación codifica la información gramatical
- El análisis de una frase forma un grafo dirigido de dependencias donde las relaciones se etiquetan con el tipo de dependencia.
- Decimos que el grafo de dependencias es proyectivo si podemos dibujar los arcos sin que se crucen.



Gramáticas de dependencias

- Las relaciones de dependencia son mucho más cercanas a las relaciones semánticas
  - No es tan fácil inferir relaciones semánticas en una estructura sintáctica
- Como las relaciones son entre palabras no es necesario aprender o representar estructuras internas de agrupamiento (sujeto, predicado, …)
- Especialmente interesantes para lenguajes donde el orden de las palabras no importa tanto

Gramáticas de dependencias



Who did what to whom?

Gramática sintáctica:

S -> NP VP
NP -> noun
VP -> V NP PP
…

Gramática de dependencias:

nsubj(NBC, suspended)
dobj(Williams, suspended)
…

Gramáticas de dependencias

- Se utilizan para representar distintos tipos de dependencias
  - Dependencias semánticas
    - Predicados con argumentos (ej. “Sara likes Sam” -> likes(Sara, Sam))
  - Dependencias mofológicas
    - Palabras con partes de palabras (género, número)
  - Prosódicas
    - Relativas a la pronunciación (unir palabras, por ej. “He’ll stop”)
  - Sintácticas

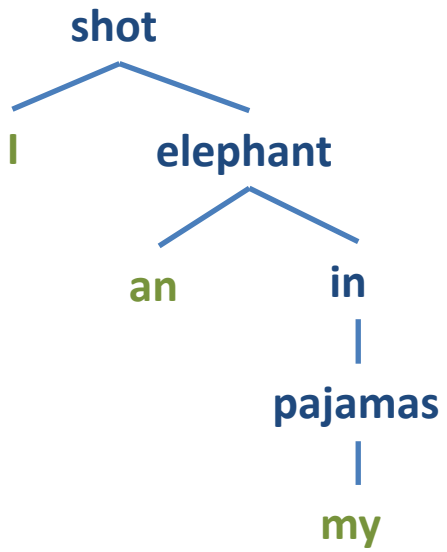
- Las relaciones sintácticas y semánticas están muy relacionadas pero no siempre coinciden

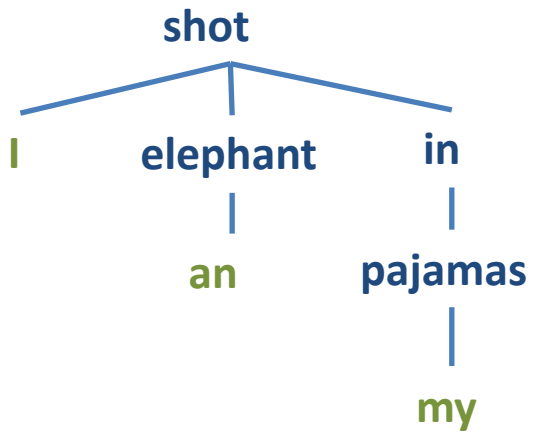
Gramáticas de dependencias

- Las gramáticas de dependencias también presentan problemas de ambigüedad

Gramática:
shot -> I
shot -> elephant
shot -> in
elephant -> an
elephant -> in
in -> pajamas
pajamas -> my

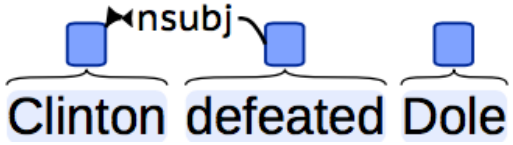
Frase: “I shot an elephant in my pajamas”

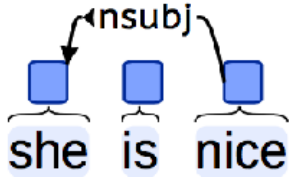




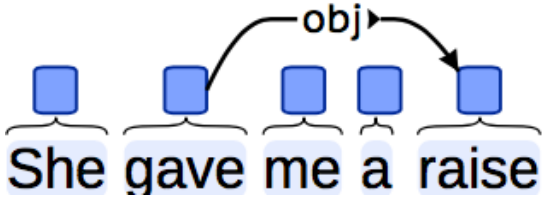
Gramáticas de dependencias

- Universal Dependencies (UD) es un framework para anotar gramáticas de forma consistente en distintas lenguas (partes del discurso, características morfológicas y dependencias sintácticas) https://universaldependencies.org/
- Ejemplos de etiquetas:
  - **nsubj**: sujeto para verbos en voz activa; cuando el verbo es copulativo el sujeto depende del complemento; …



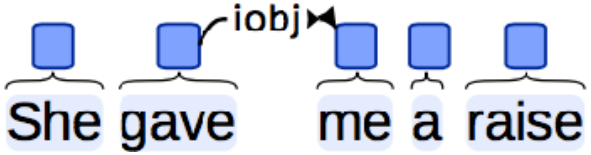


  - **obj**: generalmente el objeto directo de un verbo

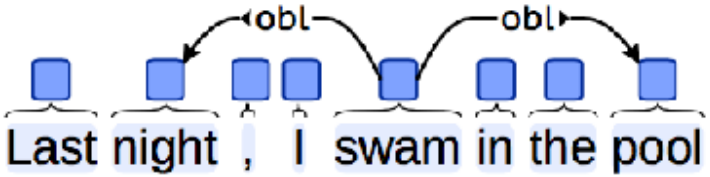


Gramáticas de dependencias

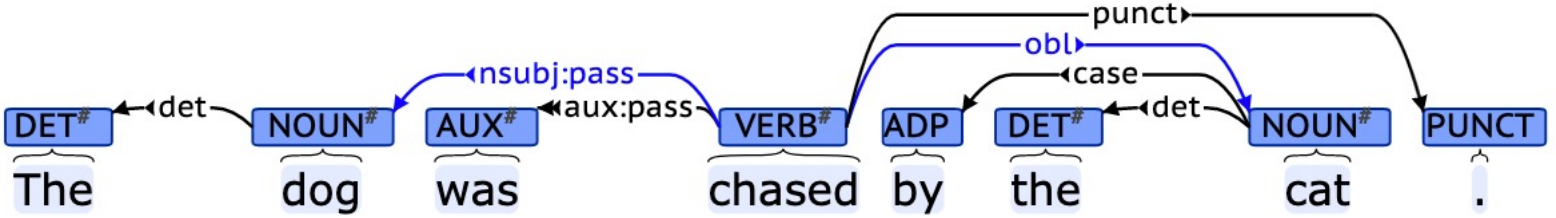
– **iobj**: generalmente el objeto indirecto de un verbo



– **obl**: otros complementos de un verbo



– Y muchas más…



Gramáticas de dependencias

- No son una gramática en el sentido de “gramática generativa”
  - No definen un lenguaje, como sí lo hacen las gramáticas independientes del contexto
  - Cualquier estructura es válida, las mejores se determinan mediante modelos probabilísticos
- Sin embargo, funciona en las tareas de análisis (*parsing*) para extraer la información clave “predicados” y “argumentos”
- Existen corpus de gramáticas de dependencias (*treebanks*) para distintos lenguajes y herramientas de búsqueda para consultarlos (lenguajes de consulta)

COMPRENSIÓN: ANÁLISIS PRAGMÁTICO

Análisis pragmático

- Análisis de los textos de forma global
  - Muchos sistemas reales no lo incluyen
- Estudiar los fenómenos que transcienden el análisis frase a frase
  - Tener en cuenta el contexto
- Clases de fenómenos del lenguaje
  - Anáfora o referencia anafórica: uso de una palabra que hace referencia o reemplaza a otras palabras (que han aparecido antes, ya sea en la misma frase o en otras)
    - En particular, hace referencia al uso de pronombres
    - Por ejemplo, en muchas frases el sujeto hace referencia a un nombre que aparece en otra frase
      - *Juan está buscando a su perro. Él piensa que lo encontrará.*

Análisis pragmático

- Tratamiento de las anáforas
  - Técnicas basadas en mantener el foco de atención
    - Los pronombres sucesivos se sustituyen por el foco de atención más cercano que concuerde sintáctica y semánticamente
    - Se mantiene la lista de entidades del discurso que se han referenciado en las últimas frases
  - Hace falta mucho conocimiento
    - *Puse el disco en el armario. Después lo cogí.*
    - *Puse el disco en el armario. Después lo cerré.*
    - *Sustituir "lo" por un nombre de la frase anterior: ¿"disco" o "armario"?*
  - Algunas referencias no son directas sino a través de *es-un* o *es-parte-de*
    - *Compré un gato. El animal no me dejaba dormir.*
    - *Compré un coche. Las ruedas estaban gastadas.*

Análisis pragmático

- Más clases de fenómenos del lenguaje
  - Elipsis: omitir partes de la frase que tienen que completarse a partir del contexto para determinar el significado
    - Por ejemplo, sujeto omitido
    - *Juan perdió su perro. Paco también.*
      - Referencia a acciones y no únicamente a cosas o personas...
  - Construcciones extra-gramaticales
    - No respetan la gramática aunque podemos darle significado
  - Construcciones metalingüísticas
    - Referencias a frases anteriores
    - *"Aunque dije que iba a llover realmente no lo parece"*
  - Intenciones de los interlocutores
    - Difícil de tratar
    - La misma frase dicha por dos personas distintas puede significar cosas distintas según la intención del que habla

Análisis pragmático

- Se necesita un conocimiento del dominio muy profundo para entender las relaciones entre distintas frases. Los programas que lo consiguen
  - o bien cuentan con unas bases de conocimiento muy grandes,
  - o bien el dominio de discurso está tan restringido que basta con una base de conocimiento más limitada
- El análisis pragmático sólo es realista si restringimos mucho el dominio
  - Sistemas de consulta a bases de datos: catálogos de pisos, ordenadores...
  - El vocabulario está predefinido y las frases tienen patrones claros
- La forma en la que se organiza el conocimiento es crítica para que los sistemas tengan éxito en esta fase

GENERACIÓN DE LENGUAJE NATURAL

Generación de lenguaje natural

- Aplicaciones en las que el lenguaje “enlatado” no es suficiente:
  - Sistemas de diálogo
  - Sistemas expertos de soporte de decisiones → generación de explicaciones
  - Tutores inteligentes
  - Sistemas de traducción automática
- Dos componentes
  - Módulo estratégico
    - Cubre la generación profunda del lenguaje
    - Se ocupa de qué decir
    - Depende totalmente de la aplicación
  - Módulo táctico
    - Generación superficial del lenguaje
    - Se ocupa de cómo decirlo

Generación de lenguaje natural

- Para la generación estratégica se tienen en cuenta:
  - Objetivos
    - Qué información se pretende transmitir al usuario
  - Planes para la consecución de estos objetivos
    - Cómo organizar y fraccionar la información que vamos a transmitir
  - Modelos de usuario
    - La información que vamos a presentar depende del usuario
    - Permiten tener en cuenta
      - Lo que sabe el usuario
      - Para qué va a usar la información el usuario
    - Estáticos o dinámicos
      - Los modelos de usuario estáticos no cambian
      - Los modelos dinámicos son aquéllos en los que el conocimiento del usuario va cambiando.
        - » Se aplican a sistemas de aprendizaje (por ejemplo, a tutores inteligentes)

Generación de lenguaje natural

- La generación táctica se compone de dos tareas:
  - Selección léxica
    - Elegir de un diccionario las palabras que se van a usar para transmitir la información
  - Selección sintáctica
    - Elegir qué tipo de frases se van a usar

Enlaces

- Enlaces interesantes:
  - **Wordnet**: A Lexical Database for English
    - https://wordnet.princeton.edu/
  - **DCGs:** Tutorial de Markus Triska
    - https://www.metalevel.at/prolog/dcg
  - **Natural Language Processing by David Bamman (UC Berkeley):** diapositivas de un curso muy completo sobre PLN
    - https://people.ischool.berkeley.edu/~dbamman/nlp20.html