

Tagging

- POS Tagging
- Taggers basados en reglas
- Taggers estadísticos
- Taggers y aprendizaje automático
- Combinación de estrategias

POS Tagging

Las palabras, tomadas en forma aislada, son ambiguas respecto a su categoría

Yo bajo con el hombre bajo a

PP	VM	SP	TD	NC	VM	NC
	VM				VM	SP
	AQ				AQ	
	NC				NC	
	SP				SP	

tocar el bajo bajo la escalera .

VM	TD	VM	VM	TD	NC	FP
VM		VM	VM	NC		
		AQ	AQ	PP		
		NC	NC			
		SP	SP			

POS Tagging

La categoría de la mayoría de las palabras no es ambigua dentro de un contexto

Yo bajo con el hombre bajo a

PP	VM	SP	TD	NC	VM	NC
	VM				VM	SP
	AQ				AQ	
	NC				NC	
	SP				SP	

tocar el bajo bajo la escalera .

VM	TD	VM	VM	TD	NC	FP
VM		VM	VM	NC		
		AQ	AQ	PP		
		NC	NC			
		SP	SP			

POS Tagging

Solución:

desambiguador morfosintáctico (Pos tagger)

El objetivo de un desambiguador (tagger) es el de asignar a cada palabra la categoría más "apropiada", dentro de un contexto

- Basados en reglas
- Estadísticos
- Híbridos

POS Tagging

W = $w_1 w_2 \dots w_n$ cadena de palabras

T = $t_1 t_2 \dots t_n$ cadena de etiquetas

Para cada palabra w_i sólo algunas de las etiquetas son válidas (excepto las palabras desconocidas). Podemos obtenerlas a partir de un diccionario o de un analizador morfológico.

$$f: W \rightarrow T = f(W)$$

Taggers basados en reglas (1)

- Utilización de conocimiento lingüístico (knowledge-driven taggers)
- Construcción (normalmente) manual de las reglas
- Número limitado (~1000) de reglas
- Modelos del lenguaje y generalizaciones establecidos de forma explícita

Taggers basados en reglas (2)

- + Reglas motivadas lingüísticamente
- + Alta precisión
 - ej. EngCG 99.5%
- Alto coste de desarrollo
- No transportable
- mayor coste de etiquetado

- TAGGIT, Green,Rubin,1971
- TOSCA, Oosdijk,1991
- Constraint Grammars, EngCG, Voutilainen,1994, Karlsson et al, 1995
- AMBILIC, de Yzaguirre et al,2000

Taggers basados en reglas (3)

- Una CG consiste en una secuencia de subgramáticas cada una de las cuales consiste en una serie de restricciones (constraints) que establecen condiciones de contexto
 - ej. (@w =0 VFIN (-1 TO))
 - descarta la categoría VFIN si la palabra anterior es “to”
- ENGCG
 - desambiguación morfosintáctica reduccionista
 - 1100 restricciones
 - 93-97% de las palabras quedan totalmente desambiguadas
 - 99.7% corrección
 - reglas heurísticas aplicables sobre el residuo
 - 2-3% ambigüedad residual con 99.6% de precisión
 - CG sintáctica
 - <http://beta.visl.sdu.dk> (demos)

Taggers estadísticos (1)

- Modelos del lenguaje adquiridos automáticamente (a partir de corpus anotado = supervisado)
- Data-driven taggers
- Uso de inferencia estadística
- Técnicas procedentes del tratamiento del habla

Taggers estadísticos (2)

- + Marco teórico bien fundamentado
- + Aproximación clara. Modelos simples.
- + Precisión aceptable
 - > 97%
- + Independencia de la lengua
- Aprendizaje del modelo
 - Sparseness
- Menor precisión

- CLAWS, Garside et al, 1987
- De Rose, 1988
- Church, 1988
- Cutting et al, 1992
- Merialdo, 1994

Taggers estadísticos (3)

- Modelos del lenguaje de tipo N-gram
 - suavizado (smoothing)
 - interpolación
- Modelos ocultos de Markov (HMM)
- Aprendizaje supervisado
 - ML (maximum likelihood)
- Aprendizaje semi-supervisado
 - Forward-Backward, Baum-Welch (EM Expectation Maximization)

Charniak, 1993
Jelinek, 1998
Manning, Schütze, 1999

Taggers estadísticos (4): N-gramas

- Las palabras se clasifican en función de sus etiquetas morfosintácticas
 - modelo 1-grama

$$P(s) = \prod_{i=1}^d P(e_i)$$

- MLE (Maximun Likelyhood Estimation) si se dispone de corpus etiquetado (desambiguado = anotado con la categoría correcta)
- EM (Expectation Maximization) si no es así
- Problema del sparseness

Taggers estadísticos (4): ejemplo N-gramas

$$\arg \max_{t_1 \cdots t_n} P(t_1, \dots, t_n \mid w_1, \dots, w_n) \approx$$

$$\prod_{k=1}^n P(t_k \mid t_{k-2}, t_{k-1}) \cdot P(w_k \mid t_k)$$

Probabilidad
contextual
(trigrams)

Probabilidad
léxica

Taggers y aprendizaje automático

- Transformaciones guiadas por los errores (Transformation-based error-driven)

Brill, 1995
Roche, Schabes, 1995

- sistema híbrido: basado en reglas adquiridas semi-automáticamente

- Máxima Entropía

- Combinación de varias fuentes de conocimiento
 - No se asume independencia
 - Admite gran número de parámetros (ej. rasgos léxicos)

Ratnaparkhi, 1998,
Rosenfeld, 1994
Ristad, 1997

Taggers y aprendizaje automático: Brill Tagger

- Basado en reglas de transformación que solucionan los errores producidos por un tagger inicial
- regla
 - cambiar la etiqueta A por la etiqueta B cuando ...
 - cada regla corresponde a la instanciación de un patrón
- patrones
 - la palabra anterior (siguiente) está etiquetada con Z
 - alguna de las dos palabras anteriores (siguientes) está etiquetada con Z
 - la palabra anterior está etiquetada con Z y la siguiente con W
 - ...
- Aprendizaje de las variables A, B, Z, W a base de un proceso iterativo que escoge en cada iteración la regla (la instanciación) que más errores corrige

Taggers y aprendizaje automático: Brill Tagger

Transformation-based Error Driven Learning

- Es un algoritmo de aprendizaje voraz “greedy” dirigido por los errores que iterativamente adquiere un conjunto de reglas de transformación
- Primeramente, se etiqueta el texto anotado (el de aprendizaje) mediante otro etiquetador (por ejemplo un HMM o la etiqueta más frecuente)
- Entonces, el algoritmo añade en cada paso la regla de transformación que mejor repara los errores actuales

Taggers y aprendizaje automático: Brill Tagger

- Se adquiere un conjunto de reglas concretas instanciando un conjunto predefinido de patrones
- Conjunción de condiciones -> transformación
- Al anotar un nuevo texto se aplican las reglas de transformación en el orden en que se han generado
- Anotador_Inicial = Etiqueta más frecuente
- Tres tipos de patrones
 - No lexicalizados
 - Lexicalizados
 - Morfológicos para tratar palabras desconocidas

Taggers y aprendizaje automático: Brill Tagger

- Patrones no lexicalizados

Change tag *a* to tag *b* when:

1. The preceding (following) word is tagged *z*.
2. The word two before (after) is tagged *z*.
3. One of the two preceding (following) words is tagged *z*.

Taggers y aprendizaje automático: Brill Tagger

1. apply initial state annotator to corpus
2. while transformations can still be found do
3. for from_tag = tag_1 to tag_n
4. for to_tag = tag_1 to tag_n
5. for corpus_position = 1 to corpus_size
6. if (correct_tag(corpus_position) == to_tag
 && current_tag(corpus_position) == from_tag)
7. num_good_transformations(tag(corpus_position - 1))++
8. else if (correct_tag(corpus_position) == from_tag
 && current_tag(corpus_position) == from_tag)
9. num_bad_transformations(tag(corpus_position - 1))++
10. find max_T (num_good_transformations(T) - num_bad_transformations(T))
11. if this is the best scoring rule found yet then store as best rule:
 Change tag from from_tag to to_tag if previous tag is T
12. apply best rule to training corpus
13. append best rule to ordered list of transformations

Taggers y aprendizaje automático: Brill Tagger

- Patrones no lexicalizados (mejores reglas adquiridas)

Change Tag			
#	From	To	Condition
1	NN	VB	Previous tag is <i>TO</i>
2	VBP	VB	One of the previous three tags is <i>MD</i>
3	NN	VB	One of the previous two tags is <i>MD</i>
4	VB	NN	One of the previous two tags is <i>DT</i>
5	VBD	VBN	One of the previous three tags is <i>VBZ</i>
6	VBN	VBD	Previous tag is <i>PRP</i>
7	VBN	VBD	Previous tag is <i>NNP</i>
8	VBD	VBN	Previous tag is <i>VBD</i>
9	VBP	VB	Previous tag is <i>TO</i>
10	POS	VBZ	Previous tag is <i>PRP</i>
11	VB	VBP	Previous tag is <i>NNS</i>
12	VBD	VBN	One of previous three tags is <i>VBP</i>
13	IN	WDT	One of next two tags is <i>VB</i>
14	VBD	VBN	One of previous two tags is <i>VB</i>
15	VB	VBP	Previous tag is <i>PRP</i>
16	IN	WDT	Next tag is <i>VBZ</i>
17	IN	DT	Next tag is <i>NN</i>
18	JJ	NNP	Next tag is <i>NNP</i>
19	IN	WDT	Next tag is <i>VBD</i>
20	JJR	RBR	Next tag is <i>JJ</i>

Taggers y aprendizaje automático: Brill Tagger

- Patrones lexicalizados

4. The current word is w and the preceding (following) word is x .
5. The current word is w and the preceding (following) word is tagged z .
6. The current word is w .
7. The preceding (following) word is w and the preceding (following) tag is t .

Taggers y aprendizaje automático: Brill Tagger

- Patrones morfológicos para tratar palabras desconocidas

Change the tag of an unknown word (from X) to Y if:

1. Deleting the prefix (suffix) x , $|x| \leq 4$, results in a word (x is any string of length 1 to 4).
2. The first (last) (1,2,3,4) characters of the word are x .
3. Adding the character string x as a prefix (suffix) results in a word ($|x| \leq 4$).

Taggers y aprendizaje automático: Brill Tagger

- Patrones morfológicos para tratar palabras desconocidas

Change the tag of an unknown word (from X) to Y if:

1. Deleting the prefix (suffix) x , $|x| \leq 4$, results in a word (x is any string of length 1 to 4).
2. The first (last) (1,2,3,4) characters of the word are x .
3. Adding the character string x as a prefix (suffix) results in a word ($|x| \leq 4$).

Taggers y aprendizaje automático: Brill Tagger

- Palabras desconocidas: mejores reglas

Change Tag			
#	From	To	Condition
1	NN	NNS	Has suffix -s
2	NN	CD	Has character .
3	NN	JJ	Has character -
4	NN	VBN	Has suffix -ed
5	NN	VBG	Has suffix -ing
6	??	RB	Has suffix -ly
7	??	JJ	Adding suffix -ly results in a word.
8	NN	CD	The word \$ can appear to the left.
9	NN	JJ	Has suffix -al
10	NN	VB	The word would can appear to the left.
11	NN	CD	Has character 0
12	NN	JJ	The word be can appear to the left.
13	NNS	JJ	Has suffix -us
14	NNS	VBZ	The word it can appear to the left.
15	NN	JJ	Has suffix -ble
16	NN	JJ	Has suffix -ic
17	NN	CD	Has character 1
18	NNS	NN	Has suffix -ss
19	??	JJ	Deleting the prefix un- results in a word
20	NN	JJ	Has suffix -ive

Taggers y aprendizaje automático: Brill Tagger

- Palabras desconocidas: mejores reglas

Change Tag			
#	From	To	Condition
1	NN	NNS	Has suffix -s
2	NN	CD	Has character .
3	NN	JJ	Has character -
4	NN	VBN	Has suffix -ed
5	NN	VBG	Has suffix -ing
6	??	RB	Has suffix -ly
7	??	JJ	Adding suffix -ly results in a word.
8	NN	CD	The word \$ can appear to the left.
9	NN	JJ	Has suffix -al
10	NN	VB	The word would can appear to the left.
11	NN	CD	Has character 0
12	NN	JJ	The word be can appear to the left.
13	NNS	JJ	Has suffix -us
14	NNS	VBZ	The word it can appear to the left.
15	NN	JJ	Has suffix -ble
16	NN	JJ	Has suffix -ic
17	NN	CD	Has character 1
18	NNS	NN	Has suffix -ss
19	??	JJ	Deleting the prefix un- results in a word
20	NN	JJ	Has suffix -ive

Taggers y aprendizaje automático: Brill Tagger

- Usando como corpus anotado 600 Kw del Wall Street
- Número de reglas de transformación: <500
- Corrección (accuracy):
 - 97.0% - 97.2% (sin palabras desconocidas)
 - La corrección de un etiquetador HMM de 3-gramas se consigue solo con 86 reglas
 - 96.6% considerando palabras desconocidas

Taggers y aprendizaje automático: Brill Tagger

- Usando como corpus anotado 600 Kw del Wall Street
- Número de reglas de transformación: <500
- Corrección (accuracy):
 - 97.0% - 97.2% (sin palabras desconocidas)
 - La corrección de un etiquetador HMM de 3-gramas se consigue solo con 86 reglas
 - 96.6% considerando palabras desconocidas

Taggers y aprendizaje automático: Brill Tagger

Transformation-based Error-driven Learning: TB(ED)L

- POS Tagging (Brill 92,95; Roche & Schabes 95; Aone & Hausman 96)
- PP-attachment disambiguation (Brill & Resnik, 1994)
- Grammar induction and Parsing (Brill, 1993)
- Context-sensitive Spelling Correction (Mangu & Brill, 1996)
- Word Sense Disambiguation (Dini et al., 1998)
- Dialogue Act Tagging (Samuel et al., 1998a,1998b)
- Semantic Role Labeling (Higgins, 2004; Williams et al., 2004; CoNLL-2004)

Taggers y aprendizaje automático: Brill Tagger

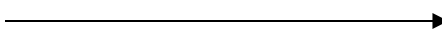
Mayores problemas de TB(ED)L

- Coste computacional
- Memoria y tiempo (especialmente en el Training)
- Algunas propuestas
 - Ramshaw & Marcus (1994)
 - LazyTBL (Samuel 98)
 - TBL (Lager 99)
 - ICA (Hepple 00)
 - FnTBL (Ngai & Florian, 01)

Taggers y aprendizaje automático

- Árboles de decisión

- Aprendizaje supervisado
 - ej. TreeTagger



Black, Magerman, 1992
Magerman 1996
Màrquez, 1999
Màrquez, Rodríguez, 1997

- Aprendizaje basado en casos
(Case-based, Memory-based Learning)

- IGTREE



TiMBL
Daelemans et al, 1996

- Aprendizaje por relajación

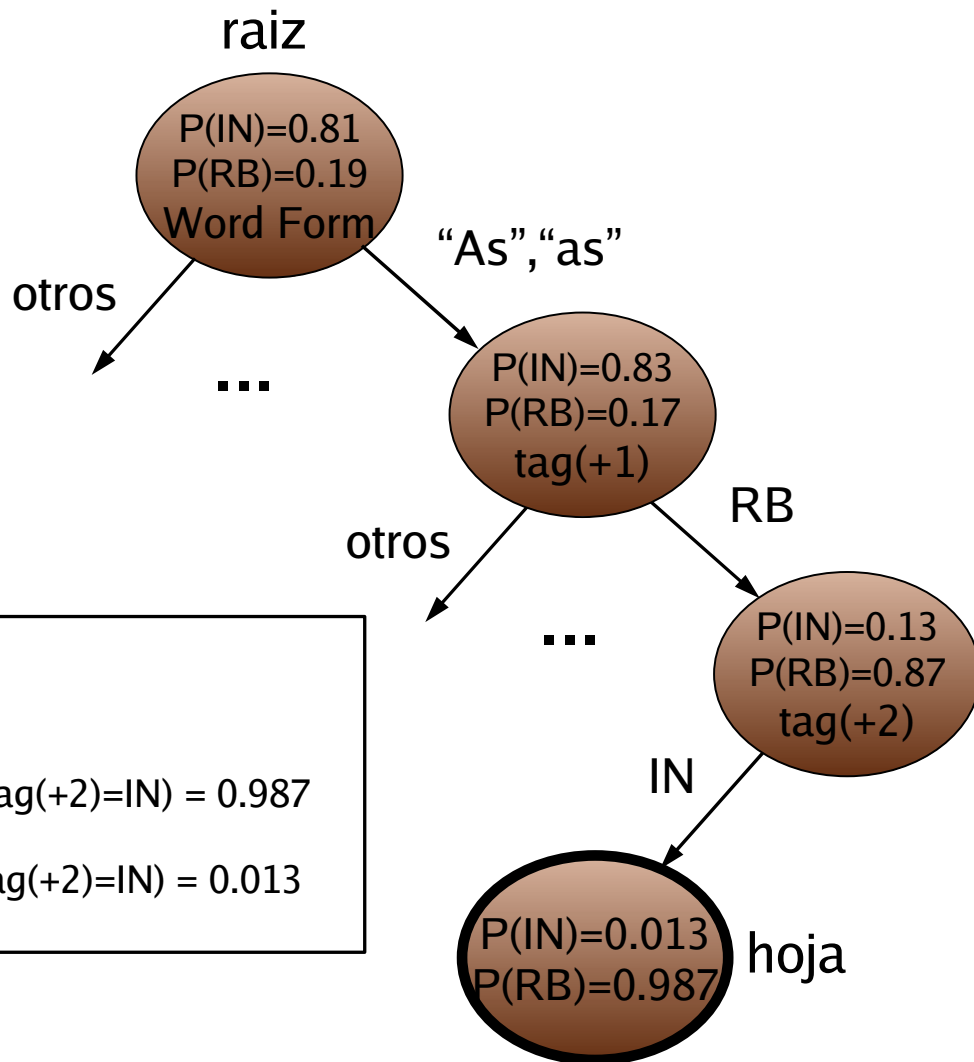
- Restricciones (constraints) de tipo estadístico o lingüístico
 - ej. RELAX



Padrò, 1997

Taggers aprendizaje automático: TreeTagger

ambigüedad IN/RB
IN (preposición)
RB (adverbio)



interpretación estadística:

$$\hat{P}(RB \mid \text{word}="A/as" \ \& \ \text{tag}(+1)=RB \ \& \ \text{tag}(+2)=IN) = 0.987$$

$$\hat{P}(IN \mid \text{word}="A/as" \ \& \ \text{tag}(+1)=RB \ \& \ \text{tag}(+2)=IN) = 0.013$$

Taggers: Combinación de estrategias

- Combinación de modelos del lenguaje en un tagger

- STT+
- RELAX

Màrquez, Rodríguez, 1998
Màrquez, 1999
Padrò, 1997

- Combinación de taggers mediante votación

- aprendizaje mediante bootstrapping

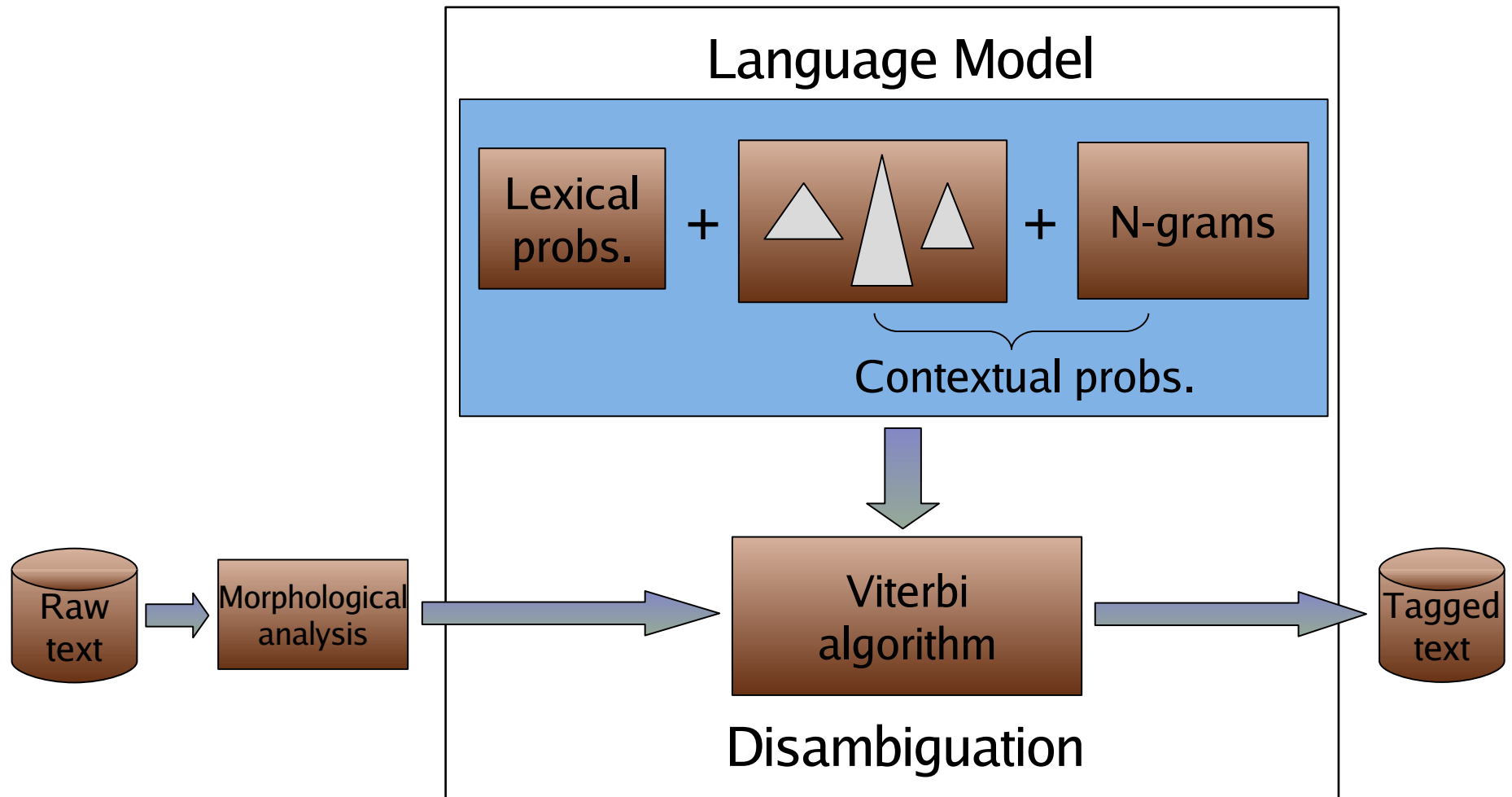
Màrquez et al, 1998

- Combinación de clasificadores

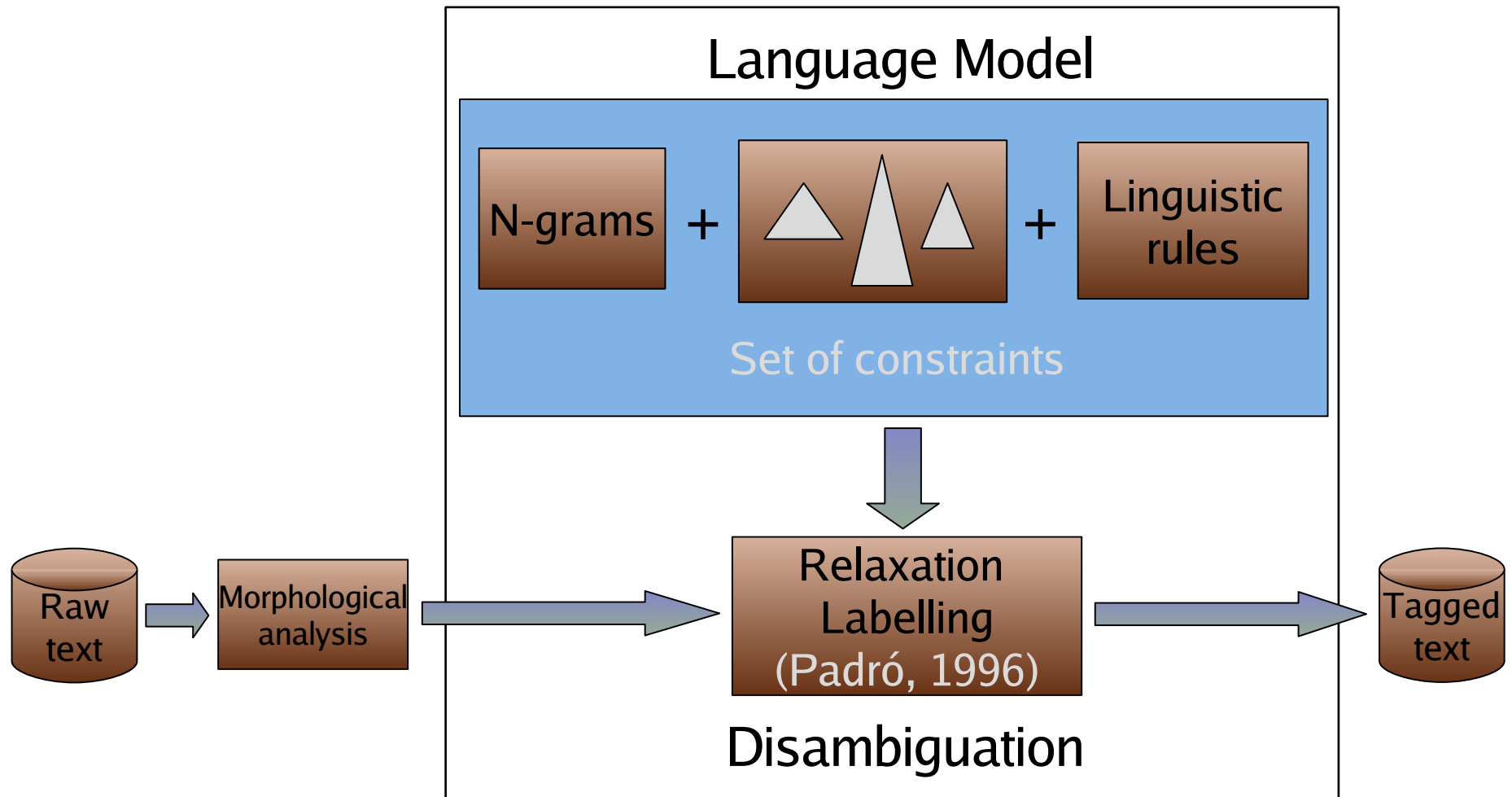
- bagging (Breiman, 1996)
- boosting (Freund, Schapire, 1996)

Brill, Wu, 1998
Màrquez et al, 1999
Abney et al, 1999

Taggers: Combinación de estrategias: STT+



Taggers: Combinación de estrategias: Relax



Medidas de calidad del etiquetado

- nivel de la oración: porcentaje de oraciones etiquetadas correctamente
 - $f(W) = \underset{T}{\operatorname{argmax}} p(T/W)$
 - Algoritmo de Viterbi (esquema de P. dinámica)
 - Más sencillo de implementar
 - Más sencillo de interpretar lingüísticamente
- nivel de la palabra: porcentaje de palabras etiquetadas correctamente
 - $f(W)^i = \underset{T}{\operatorname{argmax}} p(t_i/W)$
 - Algoritmo ML (verosimilitud máxima)
 - Puede producir secuencias no gramaticales