IIC 3800 Tópicos en CC
NLP

https://github.com/marcelomendoza/IIC3800

- DEPENDENCY PARSING -

Dependencias sintácticas

TEXT	LEMMA	POS	TAG	DEP	SHAPE	ALPHA	STOP
Apple	apple	PROPN	NNP	nsubj	Xxxxx	True	False
is	be	AUX	VBZ	aux	xx	True	True
looking	look	VERB	VBG	ROOT	xxxx	True	False
at	at	ADP	IN	prep	xx	True	True
buying	buy	VERB	VBG	pcomp	xxxx	True	False
U.K.	u.k.	PROPN	NNP	compound	X.X.	False	False
startup	startup	NOUN	NN	dobj	xxxx	True	False
for	for	ADP	IN	prep	xxx	True	True
$	$	SYM	$	quantmod	$	False	False
1	1	NUM	CD	compound	d	False	False
billion	billion	NUM	CD	pobj	xxxx	True	False

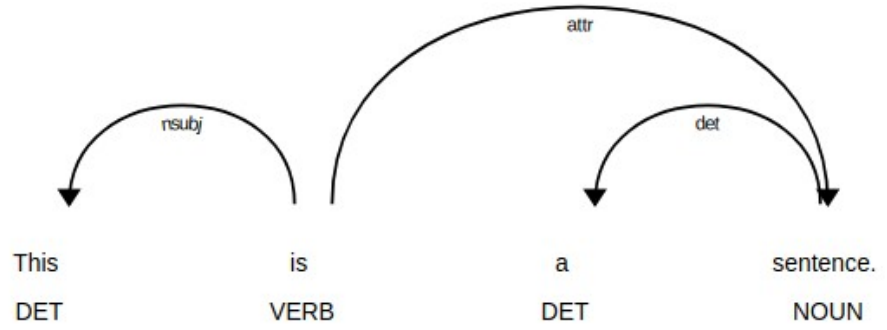
Dependencia sintáctica

Dependencias sintácticas

DEPENDENCY EXAMPLE

```
import spacy
from spacy import displacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("This is a sentence.")
displacy.serve(doc, style="dep")
```

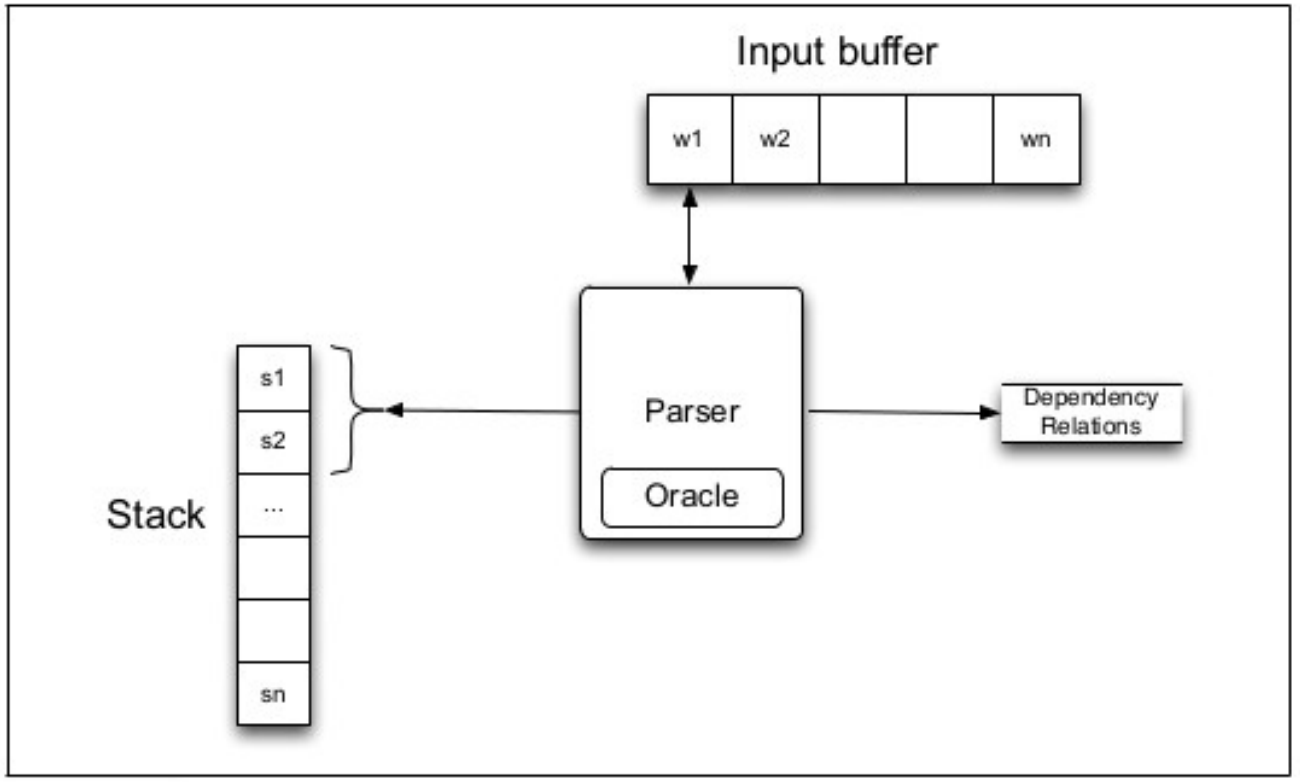


Dependencias sintácticas

Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Dependencias sintácticas

Transition-based parser:



Dependencias sintácticas

Transition-based parser: Deducimos training instances desde un treebank. Podemos determinísticamente registrar las operaciones correctas del parser, creando un training dataset.

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	

Dependencias sintácticas

Training instances (intermedia):

Actúa sobre el primer token del buffer

Stack	Word buffer	Relations
[root, canceled, flights]	[to Houston]	(canceled → United) (flights → morning) (flights → the)

La transición correcta del parser es *shift*. Luego, creamos training instances de este tipo en el dataset:

$\langle s_1.w = flights, op = shift \rangle$
$\langle s_2.w = canceled, op = shift \rangle$

← Tokens del stack

$\langle s_1.t = NNS, op = shift \rangle$
$\langle s_2.t = VBD, op = shift \rangle$

← POS-tags del stack

$\langle b_1.w = to, op = shift \rangle$
$\langle b_1.t = TO, op = shift \rangle$

← Token y tag del buffer

$\langle s_1.wt = flightsNNS, op = shift \rangle$

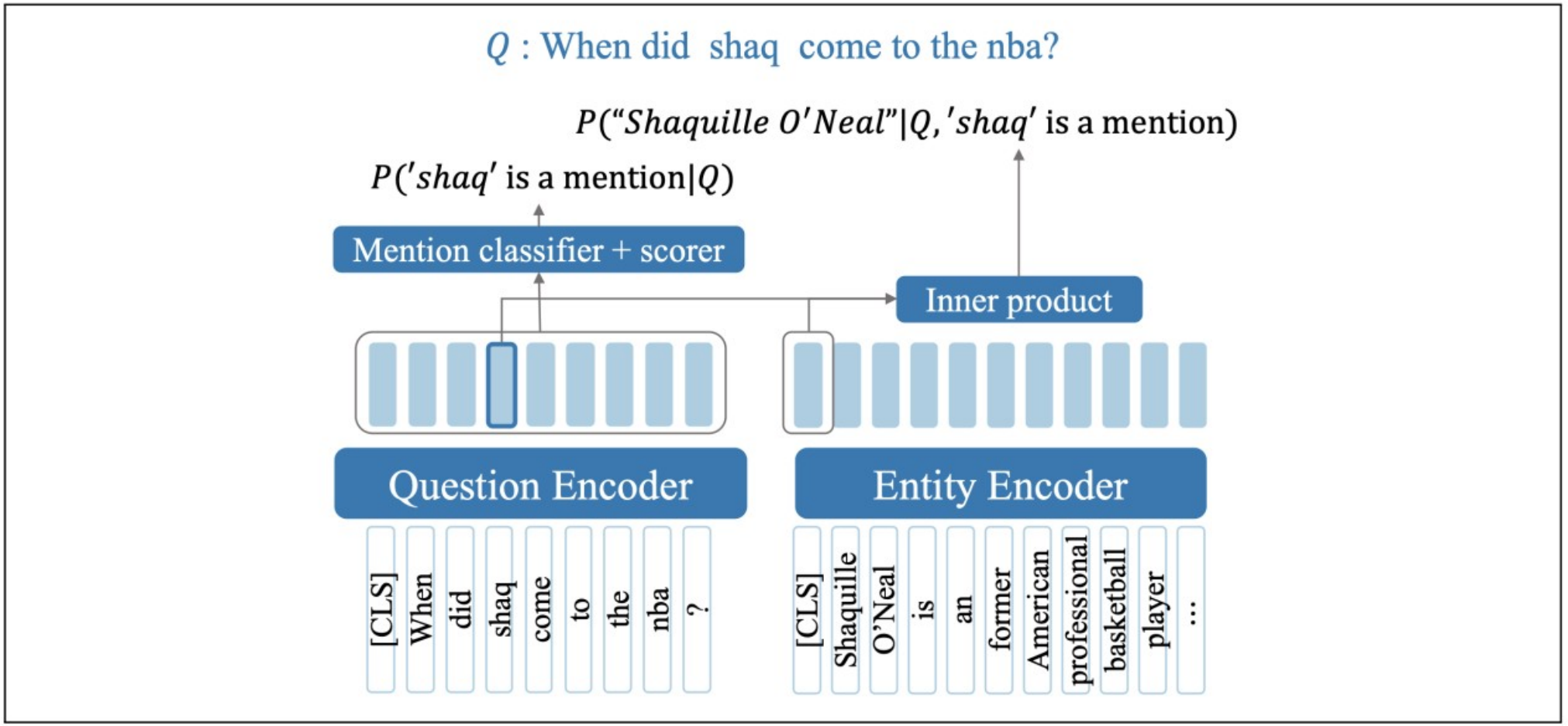
← Template compuesto

Danqi Chen, Christopher D. Manning: A Fast and Accurate Dependency Parser using Neural Networks. EMNLP2014: 740-750

- ENTITY LINKING -

Entity linking

- Conectar entidades a entidades canónicas de una KB.



Entity linking

Entity mention detection

$$[\mathbf{q}_1 \cdots \mathbf{q}_n] = \text{BERT}([\text{CLS}]q_1 \cdots q_n[\text{SEP}])$$

Embeddings de BERT por token de Q

- Se calcula la verosimilitud de cada span de Q para verificar si es una mención:

$$s_{\text{start}}(i) = \mathbf{w}_{\text{start}} \cdot \mathbf{q}_i, \qquad s_{\text{end}}(j) = \mathbf{w}_{\text{end}} \cdot \mathbf{q}_j,$$

se aprenden durante el entrenamiento

Entity linking

Entity mention detection

Luego se usa otro embedding (learnable) para detectar si un token es parte o no de la mención:

$$s_{\text{mention}}(t) = \mathbf{w}_{\text{mention}} \cdot \mathbf{q}_t$$

Las probabilidades de las menciones son obtenidas combinando estos puntajes:

$$p([i,j]) = \sigma\left(s_{\text{start}}(i) + s_{\text{end}}(j) + \sum_{t=i}^{j} s_{\text{mention}}(t)\right)$$

Entity linking

Para conectar la mención con la entidad, calculamos los embeddings de cada token que describe la entidad en la KB:

$$\mathcal{E} = e_1, \cdots, e_i, \cdots, e_w$$

$$\mathbf{x}_e = \mathrm{BERT}_{[\mathrm{CLS}]}([\mathrm{CLS}]t(e_i)[\mathrm{ENT}]d(e_i)[\mathrm{SEP}])$$

Titular Descripción

Calculamos el AWE de la mención:

$$\mathbf{y}_{i,j} = \frac{1}{(j-i+1)} \sum_{t-i}^{j} \mathbf{q}_t$$

y el producto con la entidad: $s(e,[i,j]) = \mathbf{x}_e^{\cdot} \mathbf{y}_{i,j}$

Finalmente usamos una softmax para obtener una distribución sobre la KB:

$$p(e|[i,j]) = \frac{\exp(s(e,[i,j]))}{\sum_{e' \in \mathcal{E}} \exp(s(e',[i,j]))}$$