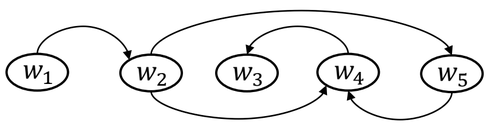
Dependency Graph Parsing as Sequence Labeling

Ana Ezquerro, David Vilares and Carlos Gómez-Rodríguez

Funded by the European Union
MINISTERIO DE CIENCIA E INNOVACIÓN
UNIÓN EUROPEA
FONDO EUROPEO DE DESARROLLO REGIONAL "Una manera de hacer Europa"
AGENCIA ESTATAL DE INVESTIGACIÓN
XUNTA DE GALICIA
citic
LyS
UNIVERSIDADE DA CORUÑA

MOTIVATION

Dependency **Graph** Parsing...



- Multiple roots and heads per node.
- Optionally non-connected.
- Cycles are allowed.

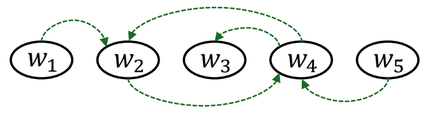
Adjacency matrix

0	0	0	0	0
1	0	0	0	0
0	0	0	1	0
0	1	0	0	1
0	1	0	0	0

Graph-based parser $O(n^2)$

Predicted scores

0	0.2	0.4	0.3	0.1
0.7	0	0	0.5	0.1
0.2	0	0	0.6	0
0.1	0.5	0.1	0	0.8
0.1	0.1	0.3	0.4	0



- Dozat & Manning (2018): ~ 90 UF
- Ji et al. (2019): ~95 UF

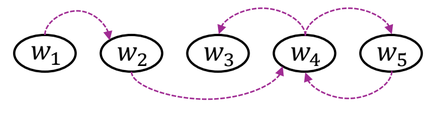
Encoding

Encoded labels: ℓ_1 ℓ_2 ℓ_3 ℓ_4 ℓ_5

SL parser $O(n)$ **Ours!**

Predicted labels: $\hat{\ell}_1$ $\hat{\ell}_2$ $\hat{\ell}_3$ $\hat{\ell}_4$ $\hat{\ell}_5$

Decoding



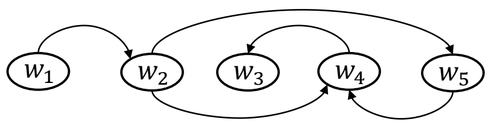
1. Define the encoding and decoding functions.
2. Then tackle as a classical tagging task.

... as **Sequence Labeling (SL)**.

UNBOUNDED ENCODINGS

Positional Encodings

- **Absolute**: Sorted sequence of head positions.
- **Relative**: Sorted sequence of head relative positions.



	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5
Absolute:	(,)	(1)	(4)	(2,5)	(2)
Relative:	(,)	(-1)	(1)	(-2, 1)	(-3)

Fast and simple decoding (easy to parallelize).

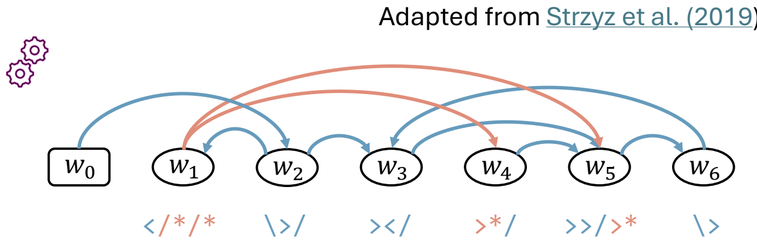
k-planar Bracketing Encoding

`<` : There is a head at the right.
`/` : There is a dependent at the right.
`\` : There is a dependent at the left.
`>` : There is a head is at the left.

- Distribute the arcs in planes (subsets of non-crossing arcs)
- Add symbol `*` for brackets corresponding to other planes.
- Second plane: `<*/*\*>*`.

- Two stacks (**L** and **R**) per plane.
- **L** for left arcs (`<\`) and **R** right arcs (`/>`).

Adapted from Strzyz et al. (2019)



After decoding ℓ_1 ...

L	1
R	0
L	
R	11

1 in **L** matches with `\` so a left arc is created $w_1 \leftarrow w_2$

$\ell_2 =$ `\>/`

0 in **R** matches with `>` so a right arc $w_0 \rightarrow w_2$ is created

BOUNDED ENCODINGS

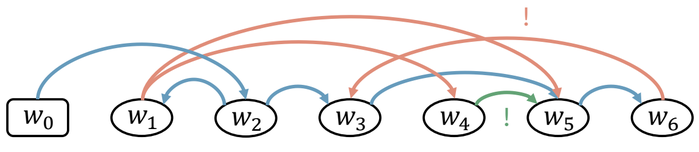
$4k$-bit Encoding

1. Distribute the arcs in relaxed 1-planar trees.
2. Encode each label ℓ_i with 4 bits: $\mathbf{b_0 b_1 b_2 b_3}$.

$\mathbf{b_0}$	**0**	w_i has a right head.
	1	w_i has a left head.
$\mathbf{b_1}$		w_i is the outermost dependent.
$\mathbf{b_2}$		w_i has left dependents.
$\mathbf{b_3}$		w_i has right dependents.

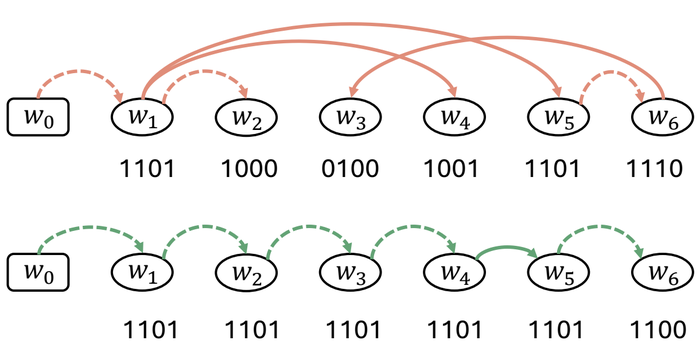
Relaxed 1-planar:
- No crossing arcs in the same direction.

Tree: Only one head per node.



... but there are nodes with no head → create artificial arcs.

- Analogous to bracketing-encoding.
- Store i and $\mathbf{b_1}$ to pop the stack.



After decoding ℓ_1 ...

L	1_1
R	0

$\ell_2 =$ 1111

$\mathbf{1_1}$ in **L** matches with $\mathbf{b_2}$=1 so a left arc is created $w_1 \leftarrow w_2$ and w_1 is the outermost, so $\mathbf{1_1}$ is removed from **L**.

0 is removed from **R** since $\mathbf{b_1}$=1

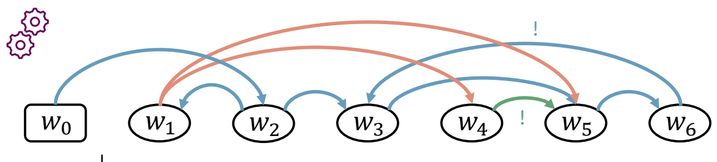
0 in **R** matches $\mathbf{b_0}$=1 so a right arc $w_0 \rightarrow w_2$ is created

Adapted from Gómez Rodríguez et al. (2023)

$6k$-bit Encoding

1. Distribute the arcs in relaxed 1-planar graphs.
2. For each node limit one head per direction.
3. Encode each label ℓ_i with 6 bits: $\mathbf{b_0 b_1 b_2 b_3 b_4 b_5}$.

$\mathbf{b_0}$	w_i has a left head.
$\mathbf{b_1}$	w_i is the outermost right dependent.
$\mathbf{b_2}$	w_i has right dependents.
$\mathbf{b_3}$	w_i has a right head.
$\mathbf{b_4}$	w_i is the outermost left dependent.
$\mathbf{b_5}$	w_i has a left dependents.

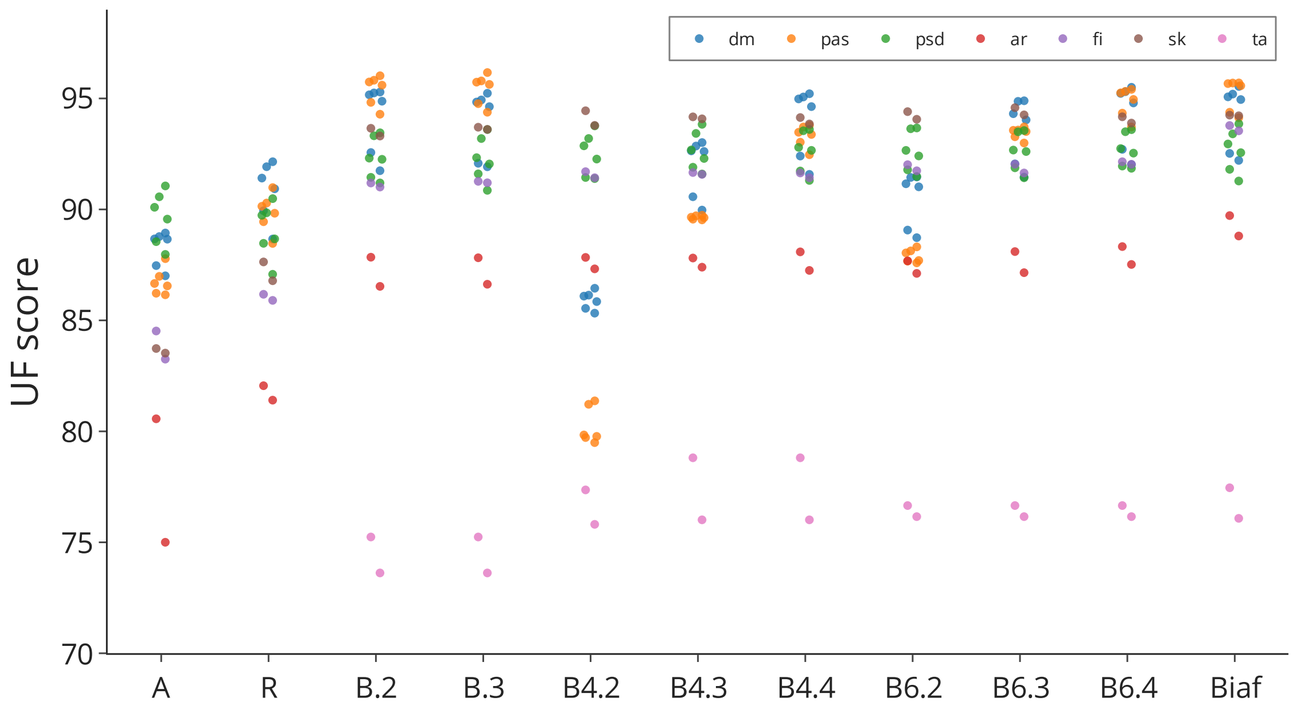


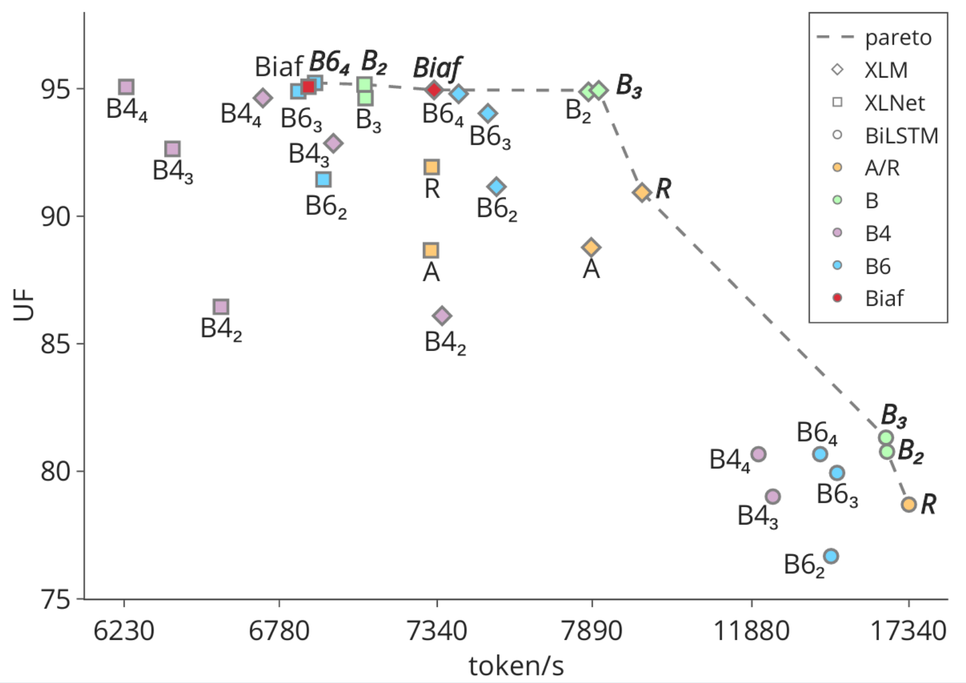
P1	000110	001001	111110	000000	111000	110001
P2	001000	000000	000000	100000	110000	000000
P3	000000	000000	000000	001000	110000	000000

EXPERIMENTS

SemEval 2015 Task 18 and IWPT-2021 Shared Task.

- BiLSTMs, XLM-RoBERTa, XLNet and $\text{FFN}_\ell + \text{FFN}_r$.
- **Biaffine** (Dozat & Manning, 2018).





We acknowledge the European Research Council (ERC), which has funded this research under the Horizon Europe research and innovation programme (SALSA, grant agreement No 101100615). We also acknowledge grants SCANNER-UDC (PID2020-113230RB-C21) funded by MICIU/AEI/10.13039/501100011033; GAP (PID2022-139308OA-I00) funded by MICIU/AEI/10.13039/501100011033/ and ERDF, EU; LATCHING (PID2023-147129OB-C21) funded by MICIU/AEI/10.13039/501100011033 and ERDF, EU; and TSI-100925-2023-1 funded by Ministry for Digital Transformation and Civil Service and "NextGenerationEU" PRTR; as well as funding by Xunta de Galicia (ED431C 2024/02), and Centro de Investigación de Galicia "CITIC", funded by the Xunta de Galicia through the collaboration agreement between the Consellería de Cultura, Educación, Formación Profesional e Universidades and the Galician universities for the reinforcement of the research centres of the Galician University System (CIGUS).