Nested Named Entity Recognition as Latent Lexicalized Constituency Parsing

Chao Lou, Songlin Yang, Kewei Tu

School of Information Science and Technology, ShanghaiTech University

Task: Nested NER

Flat NER

- Reginold Bickford, a researcher at the university of California at San Diego

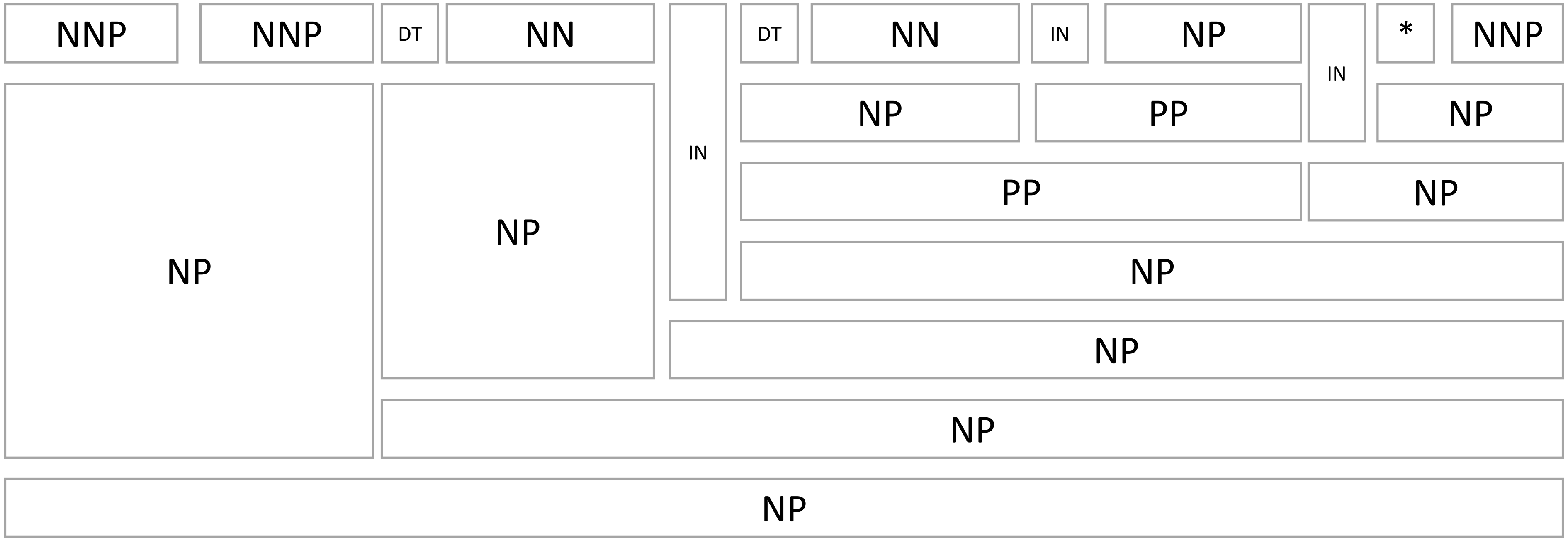
Nested NER

- Reginold Bickford, a researcher at the university of California at San Diego

Previous work: PO-TreeCRF [Fu et al., 2021]

Constituency parsing

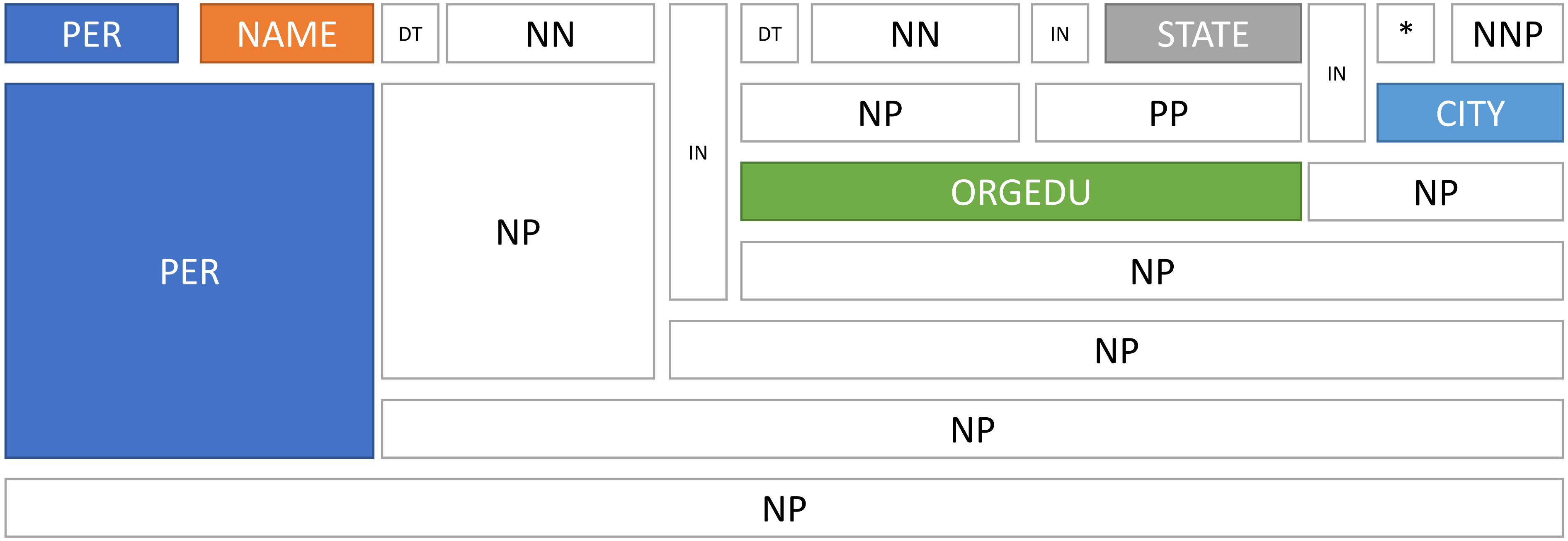
- Reginold Bickford, a researcher at the university of California at San Diego



Previous work: PO-TreeCRF [Fu et al., 2021]

Nested NER ⇋ Constituency parsing

- Reginold Bickford, a researcher at the university of California at San Diego



Previous work: PO-TreeCRF [Fu et al., 2021]

Nested NER

- Reginold Bickford, a researcher at the university of California at San Diego

Formulation: constituency parsing with **partially observed trees**

We Step Further: Lexicalization

Entity heads are important clues for entity recognition.

- Reginold Bickford, a researcher at <u>the university of California</u> at San Diego

?

?

We Step Further: Lexicalization

Entity heads are important clues for entity recognition.

- Reginold Bickford, a researcher at the university of California at San Diego

STATE

ORGEDU

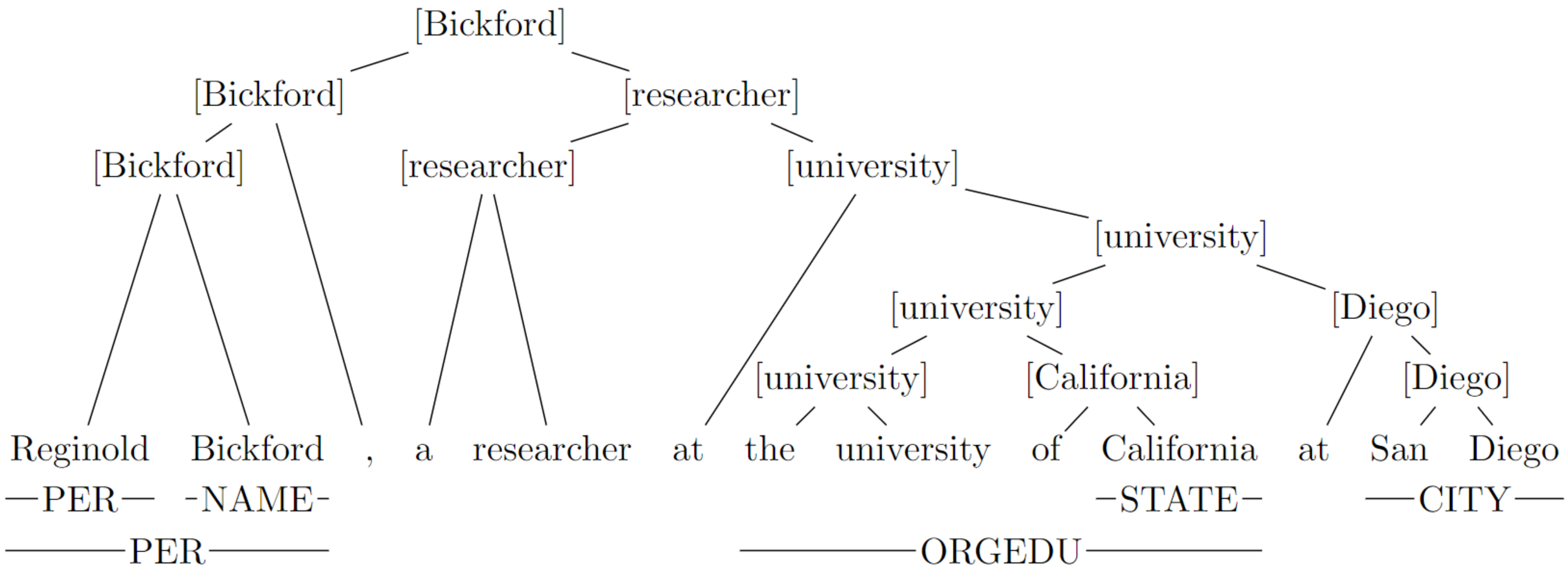
Overview

- Formulate nested NER as latent lexicalized constituency parsing
- A two-stage parsing strategy
  - Stage 1: identifying entity spans through parsing
  - Stage 2: labeling entity types
- Training loss consists of
  - a structural tree loss computed by the masked inside algorithm
  - a head regularization loss
  - a head-aware labeling loss

Our formulation: lexicalized c-parsing

- l-tree = c-tree + lexicon labels

c-tree = constituency tree
d-tree = dependency tree
l-tree = lexicalized constituency tree

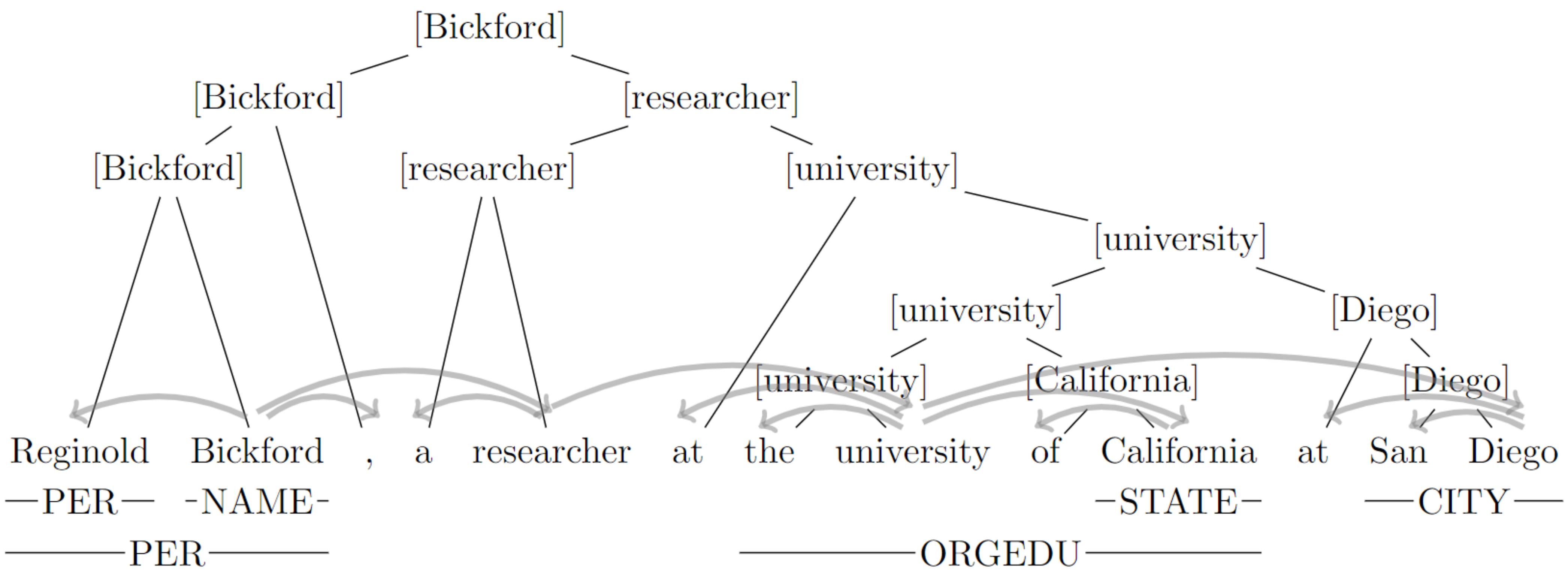


Our formulation: lexicalized c-parsing

- l-tree = c-tree + d-tree

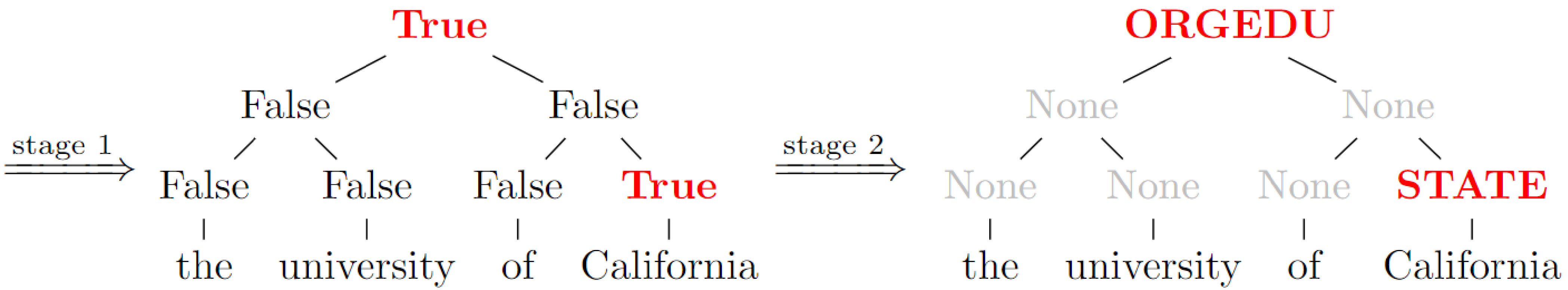
  Modeling both lexicalized spans and relations of heads

c-tree = constituency tree
d-tree = dependency tree
l-tree = lexicalized constituency tree



Our Parsing Strategy

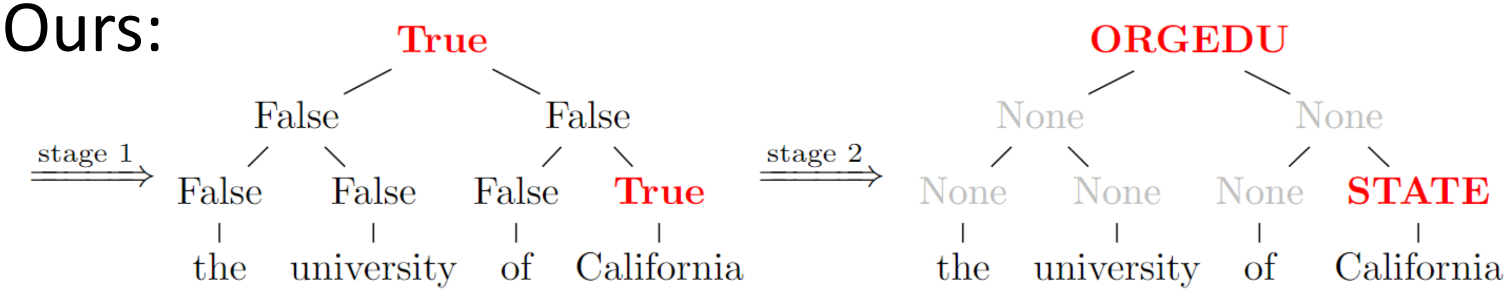
- A modified two-stage strategy
- Stage 1: predict parse trees with True/False labels
- Stage 2: predict entity labels for constituents with label True



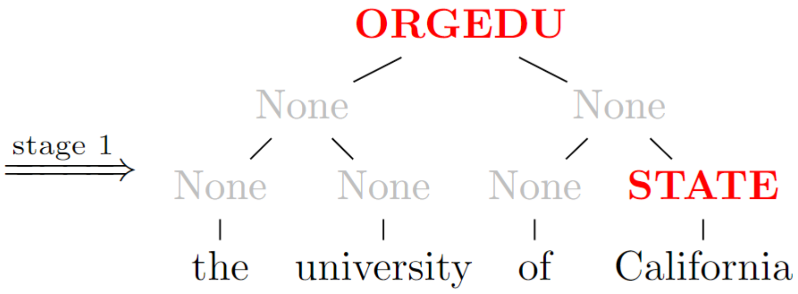
Parsing Strategy Comparison

- Ours vs. one-stage strategy

Ours:



One-stage:



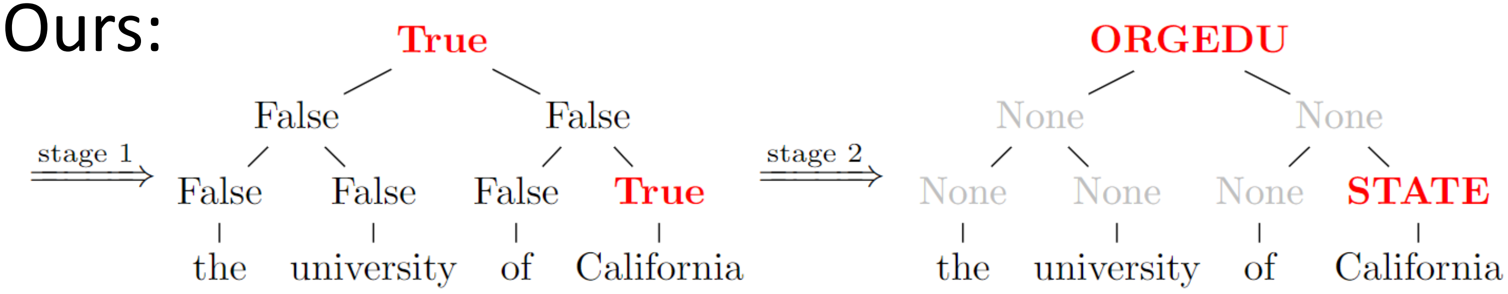
Pros:

- Support multi-label classification
- Decomposed representation for structure prediction and label prediction
- More parameters

Parsing Strategy Comparison

- Ours vs. previous two-stage strategy

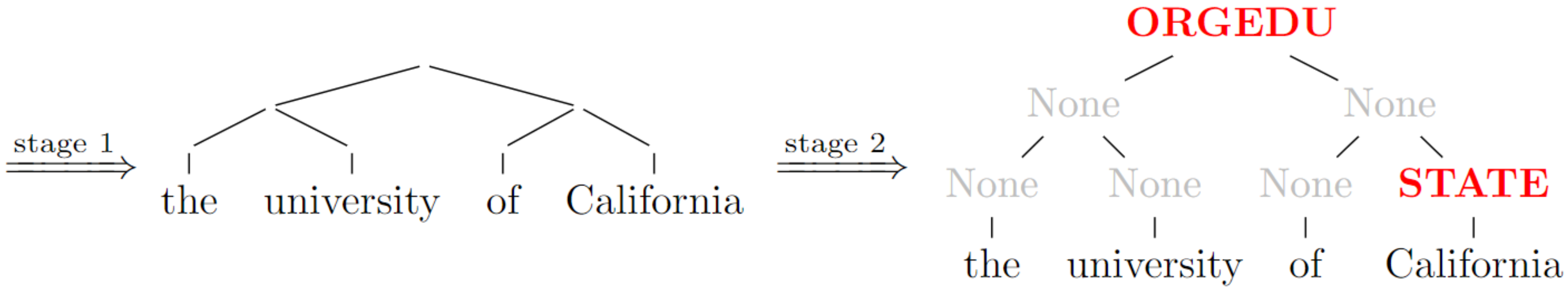
Ours:



Pros:

- Richer supervision at stage 1
- Avoid label imbalance at stage 2

Previous two-stage:



Training Loss

- Training loss =

Structural tree loss	L_{tree}
+ head regularization	L_{reg}
+ head-aware labeling loss	L_{label}

Structural tree loss L_{tree}

- Score of a l-tree is the sum of scores of spans and arcs.

$$s(l) = s(c) + s(d)$$

- Structural tree loss

$$L_{tree} = \log Z - \log \sum_{l \in \mathcal{T}} \exp\big(s(l)\big)$$

  - $\mathcal{T}$ is the set of trees containing observed entities
  - Z is the partition function
  - Use the masked inside algorithm for efficient computation of $\Sigma_{\mathcal{T}}$ [Fu et al., 2021]

Head Regularization Loss L_{reg}

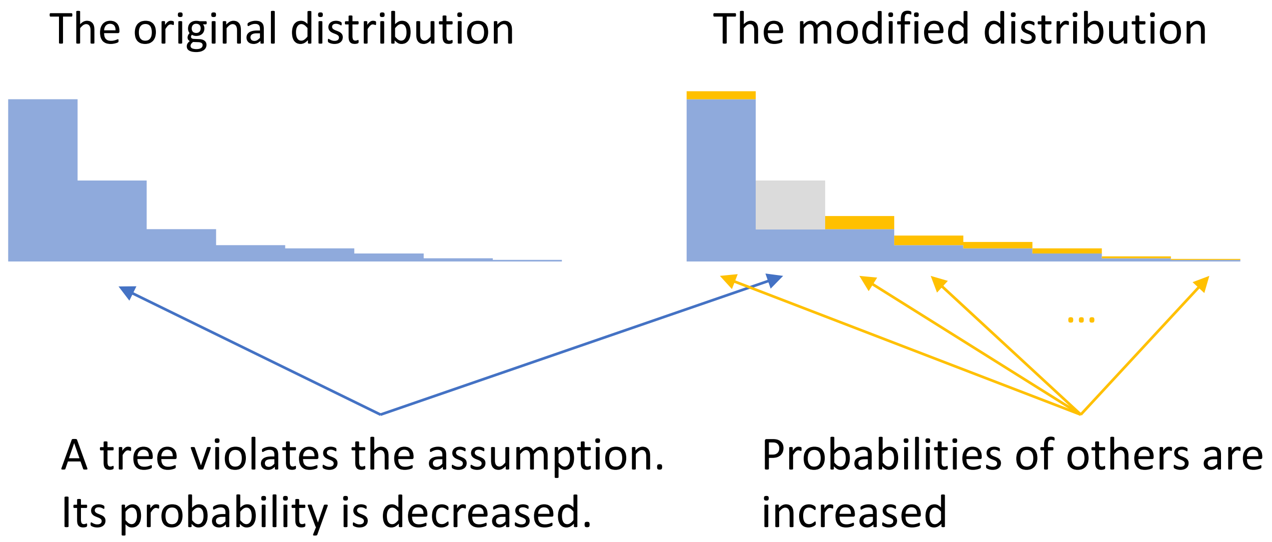
Entity heads are important clues for entity recognition.

- Reginold Bickford, a researcher at the university of California at San Diego

We prefer different heads for different entities.

Head Regularization Loss L_{reg}

- Teach model the assumption that **different entities have distinct head words**.
- Decrease the score $s(l)$ if l violates the assumption.



- Minimize the KL divergence of the two distributions.

Head-aware Labeling Loss L_{label}

- Predict labels for each span (i, j) with head k
- But we don't know the gold head
- Optimize the expected loss instead

$$L_{label} = \sum_{(i,j,y)\in N} \mathbb{E}_k L(y, \hat{y}_{ijk})$$

  - N is the set of gold entities
  - L is some loss function (e.g., cross entropy)
- Side effect: also improve the accuracy of structure prediction

Datasets

	ACE2004			ACE2005			GENIA			NNE		
	train	dev	test	train	dev	test	train	dev	test	train	dev	test
# sentences	6198	742	809	7285	968	1058	15022	1669	1855	43457	1989	3762
- nested	2718	294	388	2797	352	339	3222	328	448	28606	1292	2489
# entities	22195	2514	3034	24827	3234	3041	47006	4461	5596	248136	10463	21196
- nested	10157	1092	1417	9946	1191	1179	8382	818	1212	206618	8487	17670
- single-word	11527	1363	1553	13988	1852	1706	12933	1009	1392	166183	7291	14397
- multi-type	3	1	1	9	3	2	21	5	5	16769	792	1583

Table 9: Statistics of ACE2004, ACE2005, GENIA and NNE. An entity is considered nested if contains any entity or is contained by any entity. A sentence is considered nested if contains any nested entity.

- NNE contains lots of multi-type entities

Results

Model	ACE2004			ACE2005			GENIA		
	P	R	F1	P	R	F1	P	R	F1
Comparable									
SH	-	-	-	83.30	84.69	83.99	77.46	76.65	77.05
Pyramid-Basic	86.08	86.48	86.28	83.95	85.39	84.66	78.45	78.94	79.19
W(max)	86.27	85.09	85.68	85.28	84.15	84.71	79.20	78.16	78.67
PO-TreeCRFs†	87.62	87.57	87.60	83.34	85.67	84.49	79.10	76.53	77.80
Seq2set†	87.05	86.26	86.65	83.92	84.75	84.33	78.33	76.66	77.48
Locate&Label†	87.27	86.61	86.94	86.02	85.62	85.82	76.80	79.02	77.89
BARTNER	87.27	86.41	86.84	83.16	86.38	84.74	78.57	79.3	78.93
Ours	87.39	88.40	87.90	85.97	87.87	86.91	78.39	78.50	78.44
For reference									
SH [F]	-	-	-	83.83	84.87	84.34	77.81	76.94	77.36
Pyramid-Full [A]	87.71	87.78	87.74	85.30	87.40	86.34	-	-	-
PO-TreeCRFs [D]	86.7	86.5	86.6	84.5	86.4	85.4	78.2	78.2	78.2
Seq2set [C,P,D]	88.46	86.10	87.26	87.48	86.63	87.05	82.31	78.66	80.44
Locate&Label[C,P,D]	87.44	87.38	87.41	86.09	87.27	86.67	80.19	80.89	80.54

Table 1: Results on ACE2004, ACE2005 and GENIA. SH: Shibuya and Hovy (2020); Pyramid-Basic/Full: Wang et al. (2020)[5]; W(max/logsumexp): Wang et al. (2021)[6]; PO-TreeCRFs: Fu et al. (2020); Seq2set: Tan et al. (2021) ; Locate&Label: Shen et al. (2021); BARTNER: Yan et al. (2021). Labels in square brackets stand for the reasons of the results being incomparable to ours. F: +Flair; A: +ALBERT, C: context sentences, P: POS tags, D: different data preprocessing. † denotes that we rerun their open-sourced codes using our data.

Model	NNE		
	P	R	F1
Pyramid-Basic	93.97	94.79	94.37
Ours	94.32	94.97	94.64

Table 2: Results on NNE.

Analysis of structures

Model	P	R	F1
Unstructured(1-stage)	83.76	87.17	85.43
Unstructured(2-stage)	84.23	86.62	85.41
1-stage	84.08	87.52	85.76
1-stage + LEX	84.26	87.83	86.01
2-stage	84.68	87.33	85.99
2-stage + LEX	84.60	87.80	86.17
2-stage (0-1) + LEX	84.83	87.87	86.32
- parsing	84.26	87.40	85.83
+ head regularization	85.84	87.30	86.56
+ head-aware labeling	85.50	87.77	86.62
+ both (our final model)	**85.97**	**87.87**	**86.91**

Table 3: Ablation studies on the ACE2005 test set. LEX represents lexicalized structures.

Conclusion

- We formulate nested NER as lexicalized constituency parsing, motivated by the close relationship between entity heads and entity recognition.
- We propose a modified two-stage parsing strategy, a head regularization loss and a head-aware labeling loss to improve performance.
- The experiments on four benchmarks validate the effectiveness and efficiency of our proposed method.

Thanks!

Nested Named Entity Recognition as Latent Lexicalized Constituency Parsing