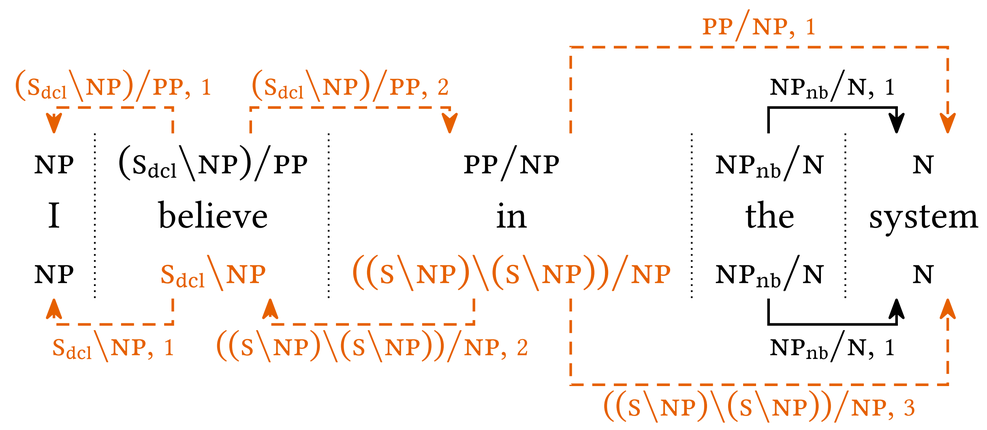
# Experts **agree**: **decomposing** CCG lexical categories **improves parser evaluations**.

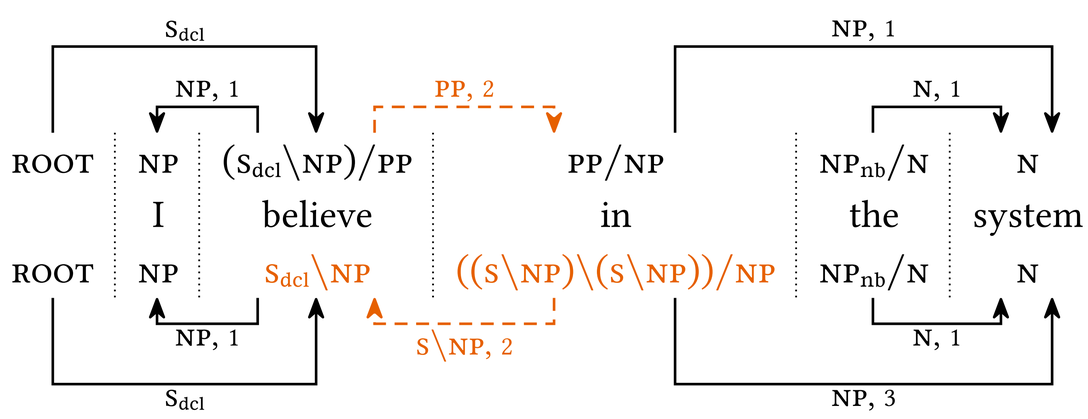
## Decomposed scoring of CCG dependencies

Aditya Bhargava and Gerald Penn

## Over-penalization in standard evaluation



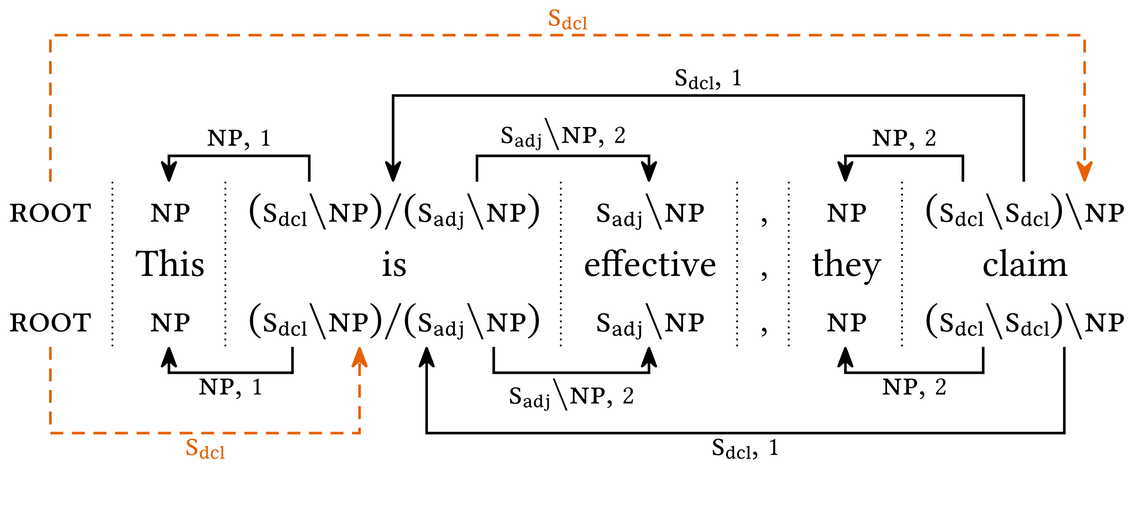

## Decomposed scoring



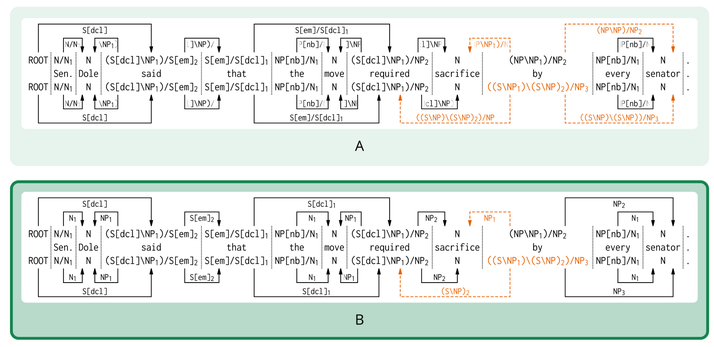

1. **Subcategorial labelling** $(S_{dcl}\backslash NP)/PP$, 1 → NP, 1 $S_{dcl}\backslash NP$, 1 → NP, 1
2. **Subcategorial alignment** Functorial Levenshtein match states:

| | ^ | S | \NP | S\NP | /NP | $ |
|---|---|---|---|---|---|---|
| ^ | 0 | 1 | 2 | 3 | 4 | 5 |
| PP | 1 | 1 | 2 | 3 | 4 | 5 |
| /NP | 2 | 2 | 2 | 3 | 3 | 4 |
| $ | 3 | 3 | 3 | 3 | 4 | 3 |

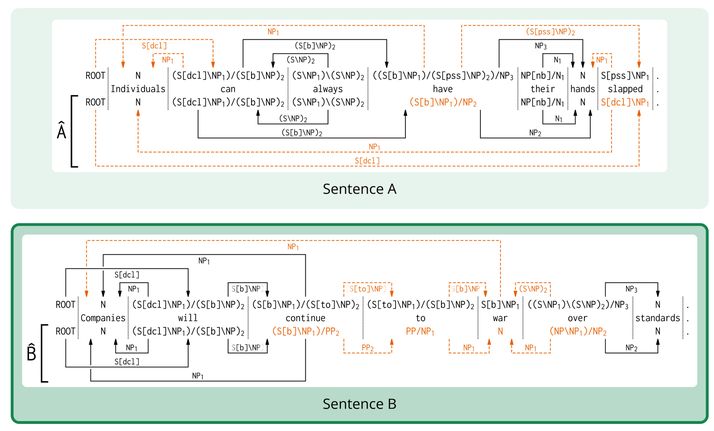
3. **Root node and dependency** Enforces *de dicto–de re* distinction:




### Intrinsic evaluation



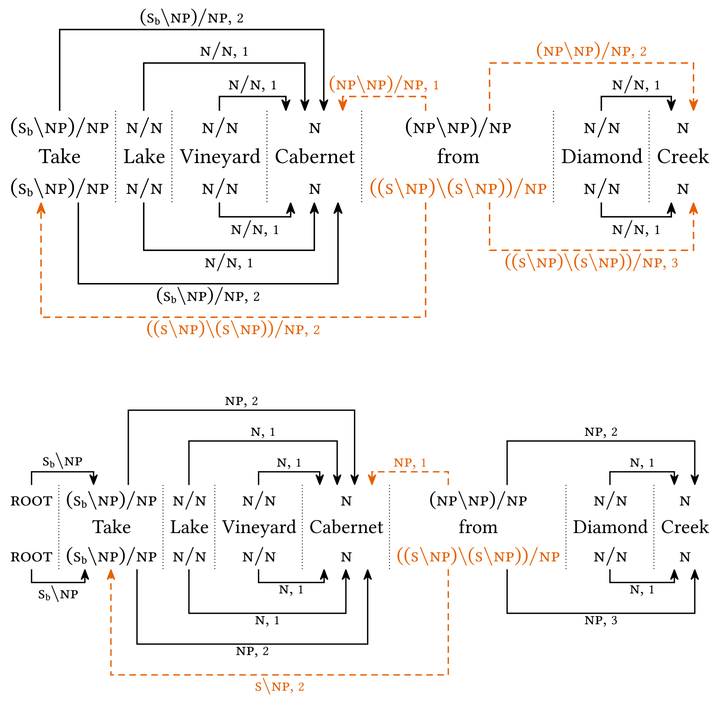

## Extrinsic evaluation




## Parser scores

| | Dev set | | Test set | |
|---|---|---|---|---|
| Parser | $F_1$ | $DF_1$ | $F_1$ | $DF_1$ |
| C&C | 83.4 | 88.4 | 84.2 | 88.9 |
| EasyCCG | 82.6 | 88.0 | 83.1 | 88.1 |
| DepCCG | 89.9 | 93.3 | 89.8 | 93.0 |

## Example: PP attachment




← Read the paper

Grab the code →