

Semantic Dependency Parsing by Bidirectional Graph-Tree Transformations and Syntactic Parsing



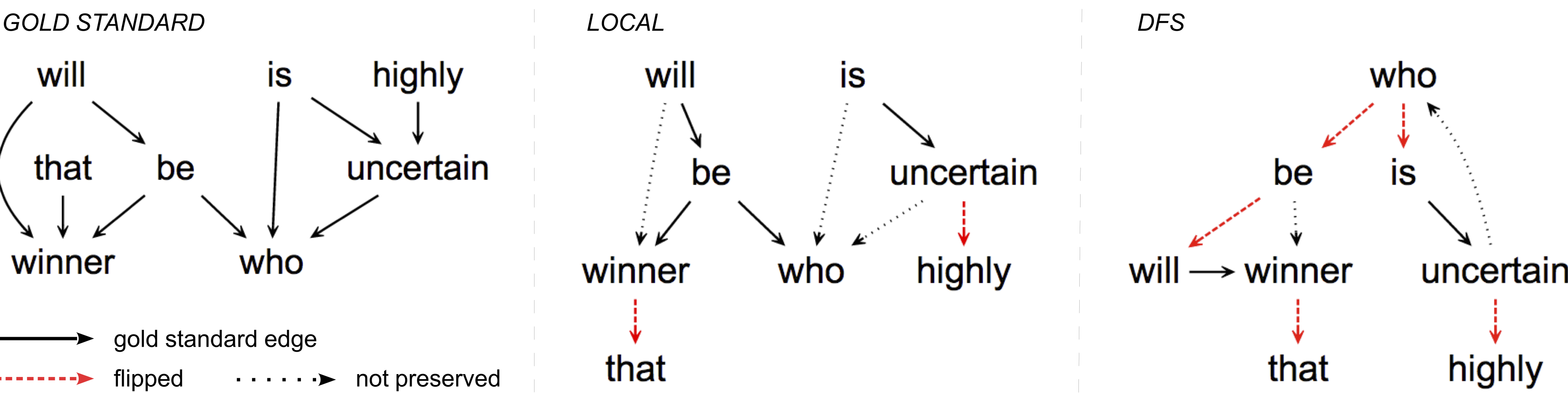
Željko Agić and Alexander Koller
Department of Linguistics, University of Potsdam, Germany
zagic@uni.potsdam.de, koller@ling.uni-potsdam.de

Introduction

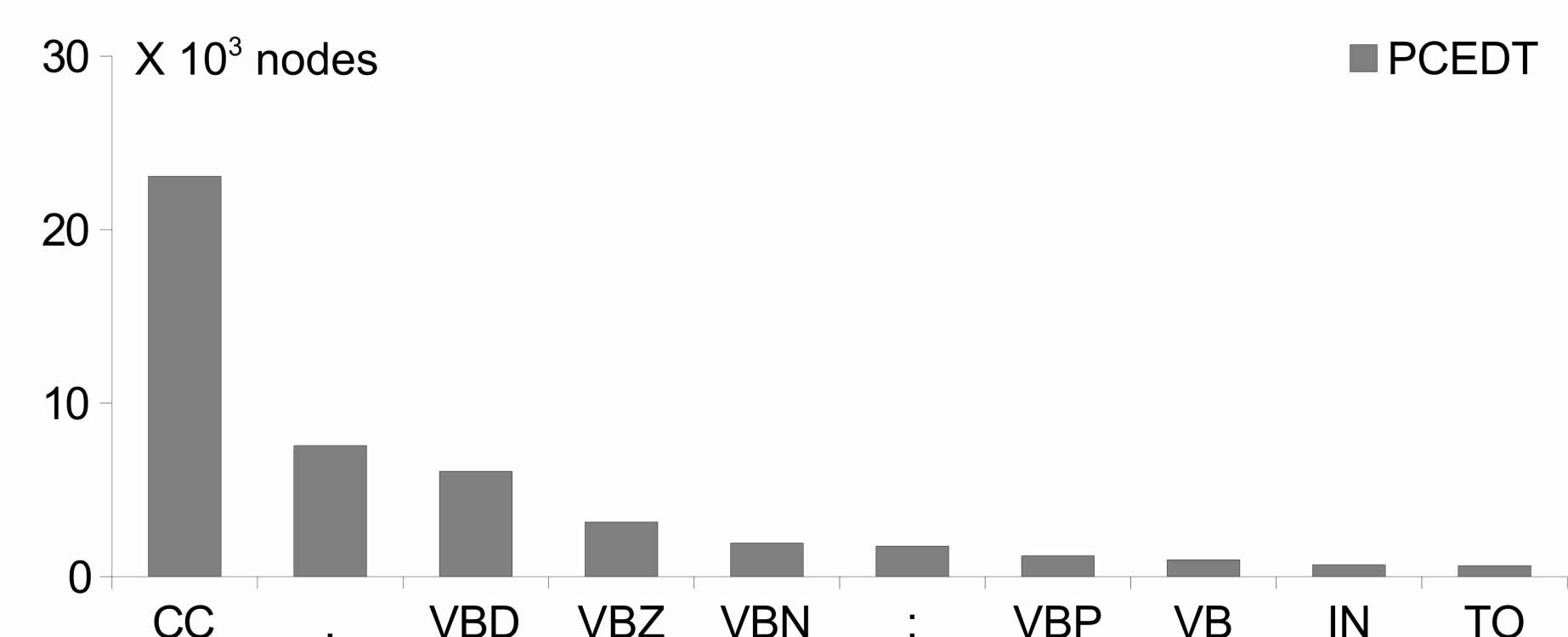
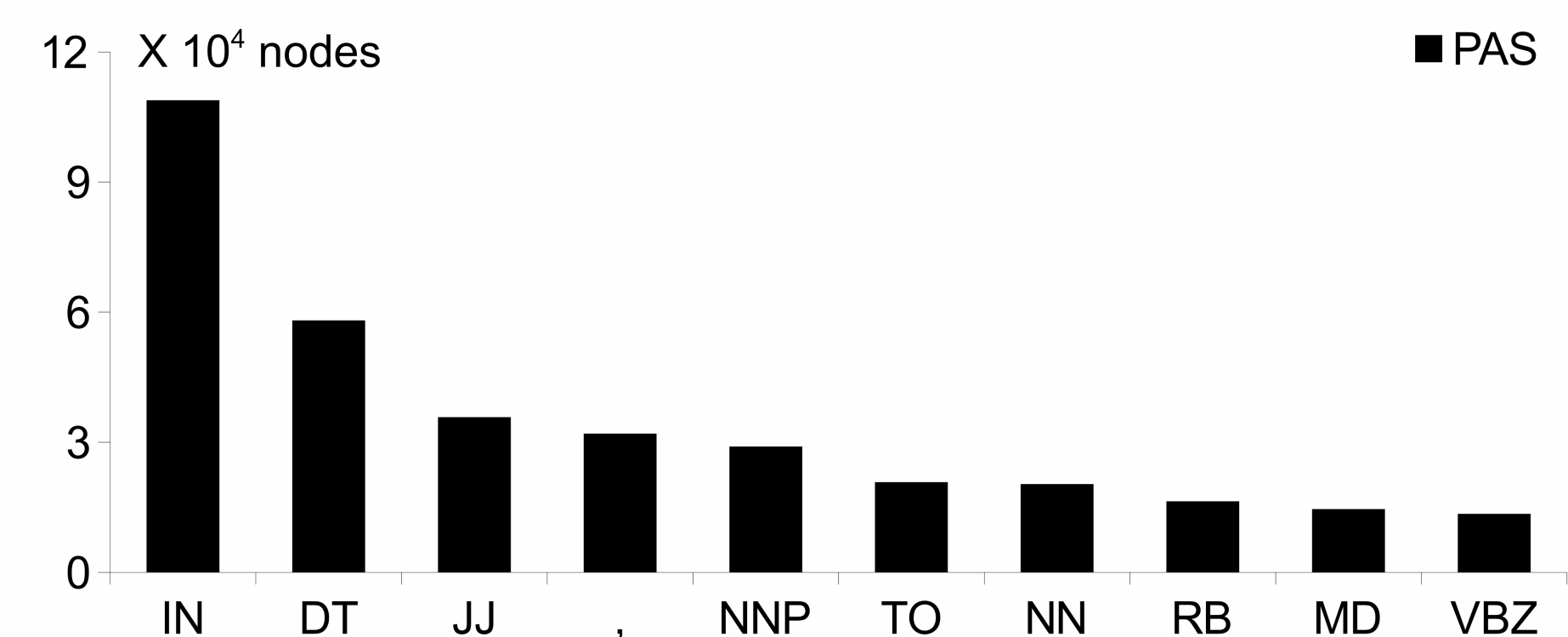
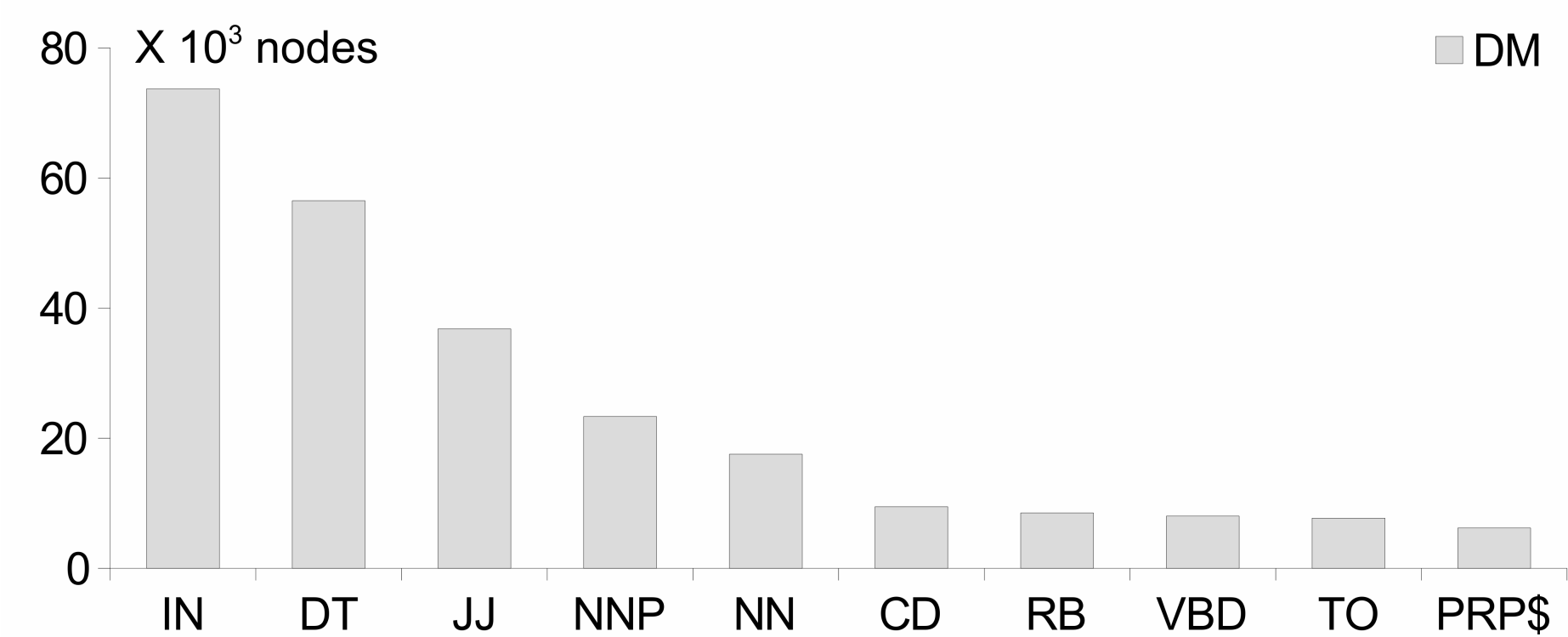
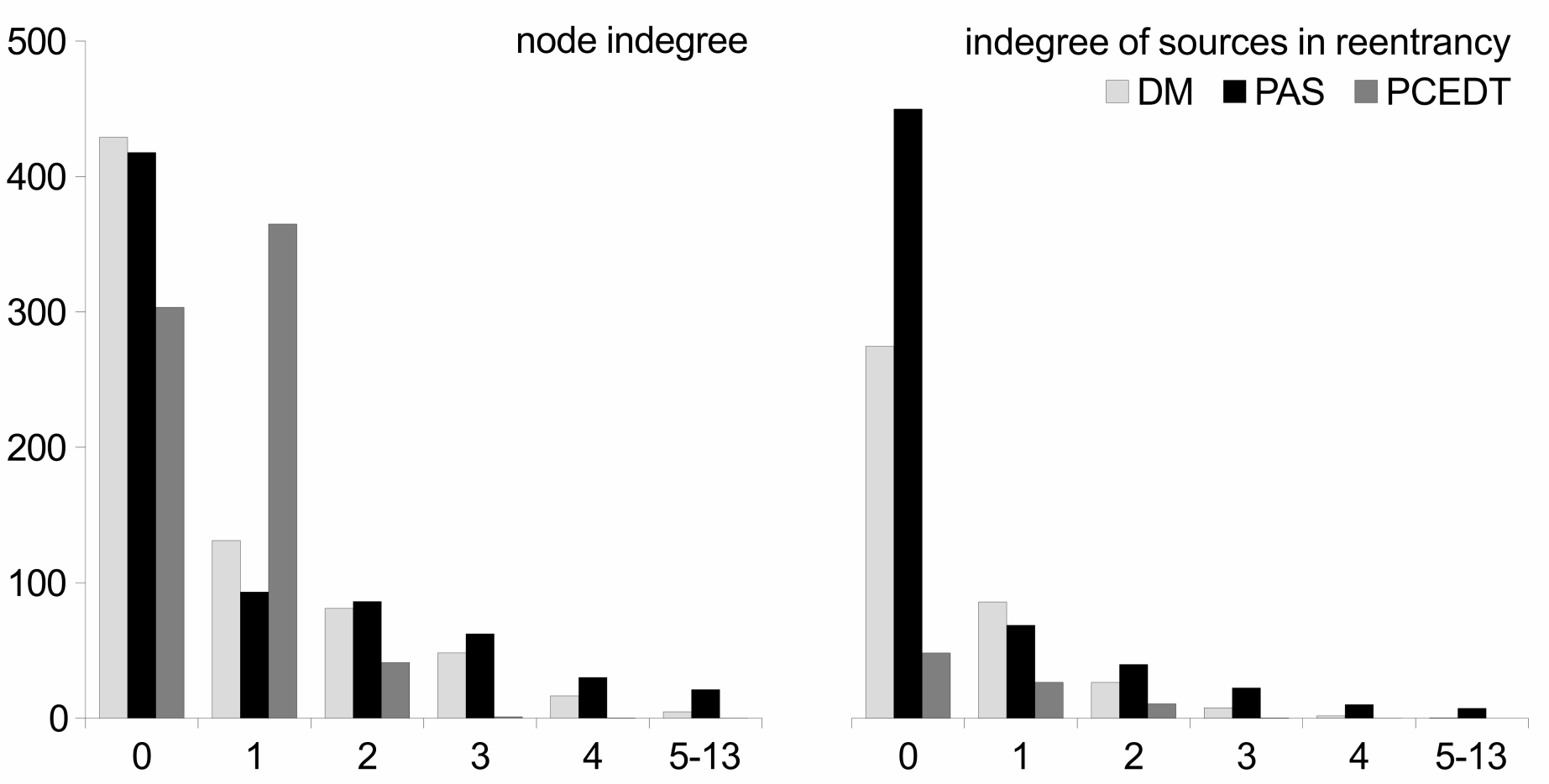
We present the [University of Potsdam](#) system from the SemEval 2014 Task 8 in data-driven [semantic dependency parsing](#). Our system is based on [transforming dependency graphs to dependency trees](#), performing tree parsing using a syntactic dependency parser, and transforming the resulting trees back to graphs. Tree [labels encode the required transformations](#).

Data and Systems

The graphs in the shared task data are relatively tree-like. We implement two transformation-based systems: *LOCAL* and *DFS*.



LOCAL is based on flipping the graph edges that end in reentrant nodes, while originating only from zero-degree nodes. *DFS* performs [depth-first graph traversal](#), and we flip all the edges following its direction. The flippings are encoded in edge labels. Official baseline is applied in post-processing. We utilize a sequence labeling top node detector. We parse using the [mate-tools graph-based dependency tree parser](#). Our systems participate in both the closed and the open track of the shared task. In the open track, we feed the companion data features to our parser at training.



Results

DFS outperforms *LOCAL* on DM and PAS data, while we marginally beat the baseline using *LOCAL* for PCEDT. Using companion data features slightly improves the overall scores. Our graph-to-tree transformations capture roughly 95% of edges in the original graphs. [Our system ranks 4th in the competition](#).

	<i>closed</i>		<i>open</i>	
	<i>LF</i>	<i>UF</i>	<i>LF</i>	<i>UF</i>
<i>LOCAL</i>	75.94	81.58	76.79	82.52
<i>DFS</i>	77.34	82.99	78.60	84.32