STARK: **A Tool for Dependency Tree Extraction and Analysis**

Kaja Dobrovoljc[1,3], Luka Krsnik[2], Marko Robnik-Šikonja[2]
[1]University of Ljubljana, Faculty of Arts
[2]University of Ljubljana, Faculty of Computer and Information Science
[3]Jozef Stefan Institute, Ljubljana, Slovenia

WG1
WG4

INTRODUCTION

- We present STARK, a recently developed tool for the **extraction of dependency trees** from Universal Dependencies treebanks.
- STARK is a python-based command-line tool, which, for a given treebank in the CONLL-U format, **produces a list of all (sub)trees** matching the various user-defined criteria, together with information on frequency and other relevant statistics.
- Through its wide selection of customizable settings, STARK facilitates **data-driven linguistic research** on various levels of grammatical description (from morphosyntactic to lexical analysis), with varying degrees of granularity (from analysis of general patterns to specific structures) and scope (from single treebank analysis to treebank comparison).
- Publicly available as an open-source software: **https://gitea.cjvt.si/lkrsnik/STARK**

CONFIGURATION SETTINGS

In addition to the **general settings**, users define the type of trees to be extracted through customizable parameters:

- **Tree size:** number of nodes as integers or range
- **Tree type:** *all* possible (sub)trees or *complete* trees only
- **Dependency type:** trees with *labeled* or *unlabeled* edges
- **Node type:** *form*, *lemma, upos, feats, deprel*
- **Node order:** *fixed* or *free*
- Additional **constraints** : label whitelist, root whitelist, specific tree query
- **Comparison** of two treebanks through the optional `compare` parameter

OUTPUT STATISTICS

Depending on the configuration, the tool returns the following types of common corpus-linguistic statistics for each tree:

- **Frequency**: number of occurrences of a tree in the input treebank (absolute and normalized)
- **Association score**: measures of the strength of association between nodes of the tree (MI, MI^3, Dice, logDice, t-score, simple-LL)
- **Keyness score**: measures for comparing patterns of frequency between the input and the reference treebank (LL, BIC, Log Ratio, Odds Ratio, % DIFF)

OUTPUT EXAMPLES

Structure	Node A	Node B	Node C	Abs. freq.	Rel. freq.	Order	Free structure	Nodes	Root
DET <det NOUN	DET	NOUN		1345	10773.0	AB	NOUN >det DET	2	NOUN
ADP <case DET <det NOUN	ADP	DET	NOUN	1163	9315.3	ABC	NOUN >case ADP >det DET	3	NOUN
ADP <case NOUN	ADP	NOUN		1090	8730.5	AB	NOUN >case ADP	2	NOUN
PRON <nmod:poss NOUN	PRON	NOUN		487	3900.7	AB	NOUN >nmod:poss PRON	2	NOUN
CCONJ <cc NOUN	CCONJ	NOUN		476	3812.6	AB	NOUN >cc CCONJ	2	NOUN

Table 1: An example output showing top-most frequent types of noun-headed trees in the English GUM Treebank (`tree_size` 2-10, `tree_type` complete, `dependency_type` labeled, `node_type` upos, `node_order` fixed, `root` upos=NOUN).

Structure	Node A	Node B	Node C	Node D	Node E	Abs. f.	Rel. f.	Order	Nodes	MI	MI3	Dice	logDice	t-score	simple-LL
Image > (: < Nick > Moreau) > .	Image	:	Nick	Moreau	.	11	88.1	ABCDE	5	37.0	43.9	0.009	7.2	3.3	223.1
On < the < other < hand > ,	On	the	other	hand	,	5	40.0	ABCDE	5	27.3	32.0	0.002	5.1	2.2	72.3
In < other < words > ,	In	other	words	,		6	48.1	ABCD	4	20.6	25.8	0.004	5.9	2.4	62.5
As < a < result > ,	As	a	result	,		5	40.1	ABCD	4	19.0	23.7	0.002	5.3	2.2	47.2
at < the < same < time	at	the	same	time		5	40.0	ABCD	4	18.3	23.0	0.003	5.7	2.2	45.2

Table 2: An example output showing top-most salient noun-headed trees in the English GUM Treebank (`tree_size` 2-10, `tree_type` complete, `dependency_type` unlabeled, `node_type` form, `node_order` fixed, `root` upos=NOUN, `frequency_threshold` 5; sorted by MI score).

VISUALISATION

STARK does not support any visualization of the output trees. However, the string describing the structure of a tree is directly transferable to the SETS treebank browsing service adopting the same dep_search query language.

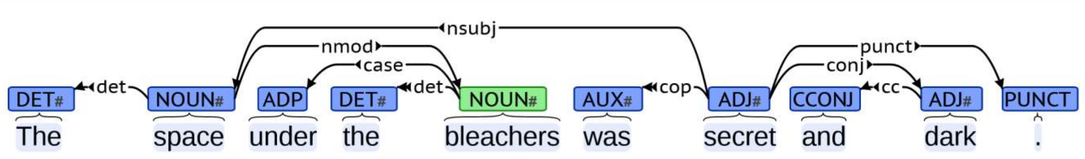

Figure 1: An example of a sentence in the English GUM Treebank featuring the `ADP <case DET <det NOUN` tree shown in Table 1.

POSSIBLE APPLICATIONS

- Using frequency-ranked lists to identify the most common / idiosyncratic lexical or grammatical patterns in a treebank.
- Using association-ranked lists to identify the most salient multi-word expressions of various types and lengths.
- Using keyness-ranked lists to identify treebank- or language-specific lexical or grammatical patterns of various kinds.