

UDON2

A PACKAGE FOR MANIPULATING UD TREES

Dmytro Kalpakchi and Johan Boye
KTH Royal Institute of Technology
Stockholm, Sweden

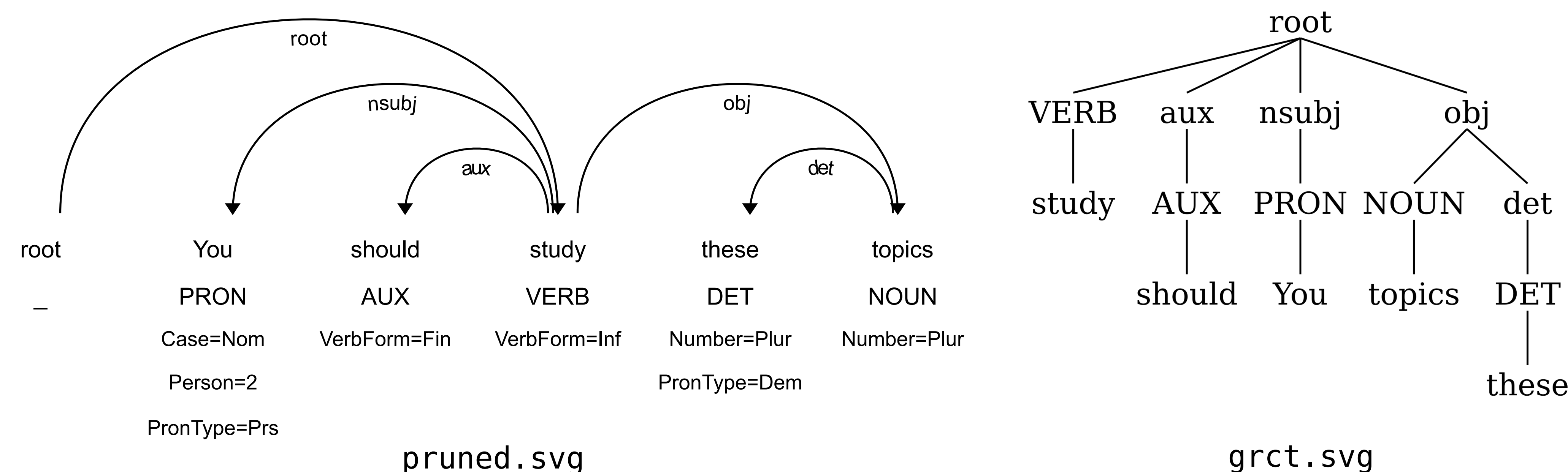


OFFERED FUNCTIONALITY

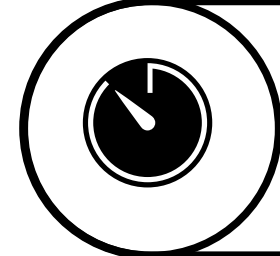
```

1 import udon2
2
3 # The only sentence: "You should study these topics or you will fail the exam"
4 nodes = udon2.ConllReader.read_file("example.conll")
5
6 r = nodes.children[0] # select the root token "study"
7 r.select_by("upos", "NOUN") # returns nodes corresponding to "topics" and "exam"
8 r.select_by_deprel_chain("obj.det") # nodes corresponding to "these" and "the"
9
10 r.prune("conj") # remove the induced subtree at the end of the "conj" edge
11 from udon2.visual import render_dep_tree
12 render_dep_tree(r, "pruned.svg") # see figure to the right
13
14 from udon2.transform import to_grct
15 render_tree(to_grct(r), "grct.svg") # see figure to the right
16
17 udon2.ConllWriter.write_to_file(nodes, "pruned.conll")

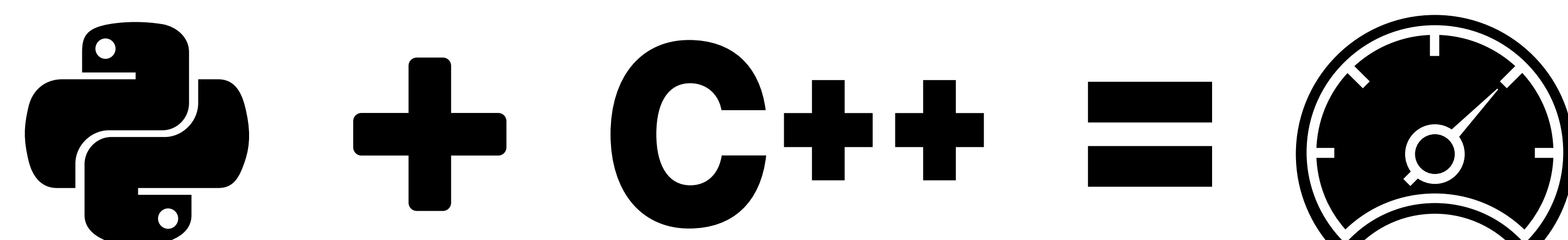
```



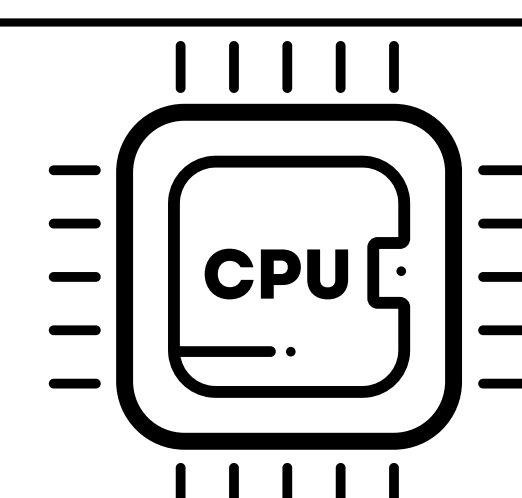
UDon2 allows transforming dependency trees to include edge labels as separate nodes into trees centered around PoS-tags (PCT), grammatical relations (GRCT) or lexicals (LCT), as introduced by Croce et al. (2011)¹. UDon2 uses these to compute **convolution tree kernels**, as proposed by Moschitti (2006)², enabling use of dependency trees in machine learning applications.



BENCHMARKS



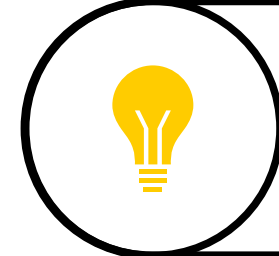
An efficient package written in C++ using Boost.Python for providing Python bindings



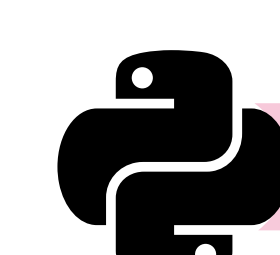
Average over 30 runs on 2.20GHz CPU

Package	OS	Memory (MiB)	Load (s)	Save (s)	Read (s)	Write (s)	Text (s)	Relchain (s)
pyconll	Ubuntu	1683.1	12.88	6.32	0.34	0.23	NA	0.47
	Windows	876.4	10.97	6.23	0.38	0.23	NA	0.54
conllu	Ubuntu	1208.7	16.83	4.28	0.19	0.1	NA	0.25
	Windows	707.2	19.11	5.23	0.22	0.09	NA	0.3
Udapi Python	Ubuntu	756.0	19.88	6.86	0.19	0.14	0.94	0.16
	Windows	421.6	19.09	8.51	0.2	0.11	1.01	0.15
UDon2	Ubuntu	772.0	3.27	3.34	0.75	0.42	0.24	0.14
	Windows	439.7	4.44	5.53	0.83	0.42	0.41	0.15

More information about benchmarks: udon2.github.io/benchmarks/

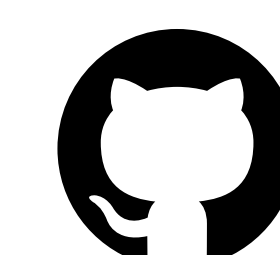


INTERESTED?



Precompiled available via PyPi for Python 3.6+ on Linux and Python 3.7+ on Windows

`pip install udon2`



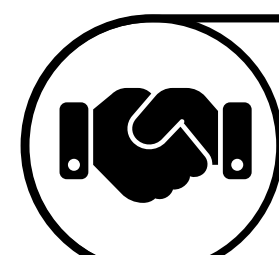
Contribute on GitHub

github.com/udon2/udon2



Documentation available online

udon2.github.io



ACKNOWLEDGEMENTS

This work was supported by Vinnova (Sweden's Innovation Agency) within project 2019-02997. We are also sincerely grateful to both reviewers for the incredibly useful comments (especially Reviewer 1 for the most thorough review we have ever seen). We would also like to thank Martin Popel for helpful discussions on the matter of benchmarking.

¹Croce, D., Moschitti, A., & Basili, R. (2011, July). Structured lexical similarity via convolution kernels on dependency trees. In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (pp. 1034-1046).

²Moschitti, A. (2006, September). Efficient convolution kernels for dependency and constituent syntactic trees. In European Conference on Machine Learning (pp. 318-329). Springer, Berlin, Heidelberg.