Are UD Treebanks Getting More Consistent? A Report Card for English UD

AMIR ZELDES & NATHAN SCHNEIDER

{amir.zeldes,nathan.schneider}@georgetown.edu

GEORGETOWN UNIVERSITY

GURT 2023 SYNTAXFEST
Washington, D.C.
Sixth Workshop on Universal Dependencies (UDW2023)

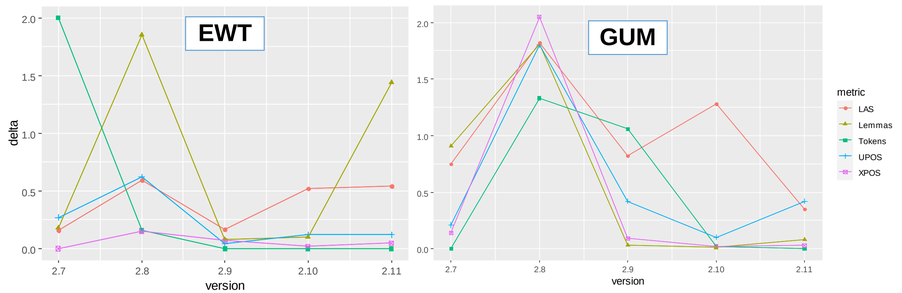
Overview

- Universal Dependencies (UD) provides 200+ treebanks in 138 languages with a unified scheme (de Marneffe et al. 2021)
- 40/138 languages have **multiple treebanks,** allowing **joint models**
- **English** default for popular tools uses EWT+GUM (Stanza, Qi et al. 2020)
- But treebanks are not necessarily **consistent** and constantly **changing**
  - ➢ How **consistent** are English EWT and GUM? Where do they differ?
  - ➢ Is consistency **improving** across UD versions? (focus on v2.6-2.12)
  - ➢ Is **joint training** for English a good idea? If so, since when?

PAPER

How has the data changed?

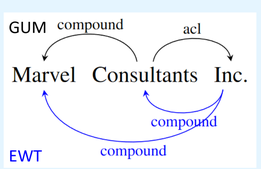
- Methodology: treat each successive version as gold and the previous as pred
- Use official CoNLL scorer to obtain delta to next version to quantify change

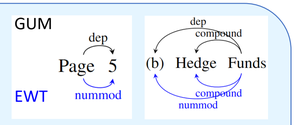


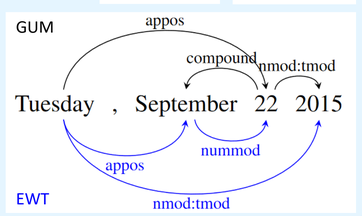
- Introduced MWTs in v2.7 & 2.8
- upos changes to proper names (**ADJ**, **VERB** in names) in 2.8
- lemma caps consistency in 2.8
- LAS due to: amod in names (2.8); parataxis for *X so Y* (2.10); nested subjects (**nsubj:outer**), relatives, clefts (2.11)
- Introduced MWTs in 2.8, split hyphenated tokens in 2.9
- xpos added **HYPH**, removed **-LSB-** to match EWT in 2.8-2.9
- **PRON** & **DET** revisions in 2.11
- named lemma consistency (2.8)
- LAS: changes to **flat** (2.10), less **dep**, addition of **orphan** cases (2.8, 2.10)

Where do GUM and EWT trees still differ?

- Proper name internal structure (incl. conversion errors)
- Some compounds
- Number modifiers
- List markers (LS)
- Dates
- Deprel *list* (almost unused in GUM)







Parsing experiments

- **Is cross-corpus parsing getting better?**
  - Methodology: fix GUM train to 2.6 documents (GUM has grown since)
  - Use Diaparser (Attardi et al. 2021) + Electra (Clark et al. 2020)

		EWT test		GUM test		Macro-Avg	
train	**version**	**UAS (sd)**	**LAS (sd)**	**UAS (sd)**	**LAS (sd)**	**UAS (sd)**	**LAS (sd)**
EWT	v2.6	92.82 0.132	90.24 0.066	87.81 0.073	83.89 0.023	90.31 0.059	87.07 0.025
	v2.7	92.84 0.037	90.25 0.173	87.87 0.088	84.19 0.074	90.35 0.062	87.22 0.114
	v2.8	92.93 0.060	90.42 0.090	87.97 0.078	84.90 0.028	90.45 0.065	87.66 0.042
	v2.9	92.88 0.107	90.41 0.131	87.57 0.148	84.36 0.105	90.23 0.098	87.38 0.117
	v2.10	93.06 0.082	90.70 0.158	87.81 0.084	84.72 0.138	90.44 0.082	87.71 0.088
	v2.11	**93.18** 0.142	**90.90** 0.139	**88.05** 0.260	**84.74** 0.289	**90.62** 0.196	**87.82** 0.207
GUM	v2.6	86.53 0.357	81.78 0.397	91.37 0.201	87.90 0.141	88.95 0.187	84.84 0.209
	v2.7	86.69 0.336	82.28 0.322	91.66 0.156	88.24 0.284	89.18 0.242	85.26 0.299
	v2.8	87.02 0.133	82.90 0.214	91.88 0.132	88.86 0.159	89.45 0.002	85.88 0.041
	v2.9	87.42 0.143	83.43 0.025	91.88 0.300	88.78 0.281	89.65 0.219	86.11 0.140
	v2.10	87.53 0.190	83.79 0.191	92.16 0.216	89.24 0.191	89.85 0.203	86.51 0.191
	v2.11	**88.23** 0.198	**84.27** 0.095	**92.28** 0.137	**89.48** 0.224	**90.26** 0.121	**86.88** 0.132

Table 1: Cross-corpus parsing scores (three run averages with standard deviations)

- Cross-corpus results **are** getting better but are **worse** than within-corpus
- GUM is **harder** (-1.4 LAS at best); 12 genres incl. spoken, less data

- **Are joint models a good idea?**
  - Two settings: GUM 2.6 documents (for fairness) or all GUM in each version (=realistic, what you get e.g. in Stanza)

		EWT test		GUM test		Macro-Avg	
train	**version**	**UAS (sd)**	**LAS (sd)**	**UAS (sd)**	**LAS (sd)**	**UAS (sd)**	**LAS (sd)**
JOINT$_{subset}$	v2.6	92.38 0.044	89.59 0.108	90.08 0.366	86.80 0.326	91.23 0.177	88.20 0.146
	v2.7	92.31 0.078	89.61 0.072	90.15 0.311	86.96 0.360	91.23 0.122	88.29 0.148
	v2.8	92.49 0.159	89.99 0.128	90.51 0.351	87.86 0.449	91.50 0.154	88.92 0.195
	v2.9	92.39 0.324	89.80 0.278	90.63 0.392	87.91 0.415	91.51 0.086	88.85 0.114
	v2.10	92.62 0.034	90.24 0.058	90.51 0.418	87.86 0.381	91.56 0.192	89.05 0.163
	v2.11	**92.92** 0.072	**90.58** 0.052	**90.75** 0.073	**87.94** 0.059	**91.83** 0.064	**89.26** 0.045
JOINT$_{all}$	v2.6	92.38 0.044	89.59 0.108	90.08 0.366	86.80 0.326	91.23 0.177	88.20 0.146
	v2.7	92.31 0.078	89.61 0.072	90.15 0.311	86.96 0.360	91.23 0.122	88.29 0.148
	v2.8	92.07 0.277	89.55 0.312	91.26 0.267	88.72 0.247	91.66 0.077	89.14 0.066
	v2.9	92.27 0.154	89.77 0.287	90.81 0.084	88.12 0.123	91.54 0.110	88.95 0.176
	v2.10	92.18 0.018	89.86 0.010	91.54 0.170	88.99 0.211	91.86 0.092	89.43 0.110
	v2.11	**92.54** 0.259	**90.11** 0.240	**91.71** 0.426	**89.11** 0.534	**92.13** 0.147	**89.61** 0.181

Table 2: Joint training parsing scores (three run averages with standard deviations)

- Scores for joint model have gotten steadily better
- Still can't beat train/test on **single** same corpus!
- But macro-average on both corpora is much better
- And the gap is now **very small** even within-corpus (best joint model less than -0.5 LAS away from best within-corpus model for both corpora)

Bottom line: in realistic usage on new data **use joint models!!**

What do parsers get wrong? (more in paper!)