Overview of the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies

Bouma, Gosse; Seddah, Djame; Zeman, Dan

Overview of the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies

Gosse Bouma* **Djamé Seddah†** **Daniel Zeman°**

*University of Groningen, Centre for Language and Cognition
†INRIA Paris
°Charles University in Prague, Faculty of Mathematics and Physics, ÚFAL
g.bouma@rug.nl, djame.seddah@inria.fr
zeman@ufal.mff.cuni.cz

Abstract

This overview introduces the task of parsing into enhanced universal dependencies, describes the datasets used for training and evaluation, and evaluation metrics. We outline various approaches and discuss the results of the shared task.

1 Introduction

Universal Dependencies (UD) (Nivre et al., 2020) is a framework for cross-linguistically consistent treebank annotation that has so far been applied to over 90 languages. UD defines two levels of annotation, the basic trees and the enhanced graphs (EUD).

In 2017 (Zeman et al., 2017) and 2018 (Zeman et al., 2018) there were CoNLL shared tasks on multilingual UD parsing that attracted a substantial number of participants. While the previous tasks evaluated morphology and prediction of basic dependencies on the UD data, the current task's focus is on predicting enhanced dependency representations. The evaluation was done on datasets covering 17 languages from four language families. The current task was organized as a part of the 16th International Conference on Parsing Technologies[1] (IWPT), collocated with ACL 2020, as a follow-up to stimulate research on parsing natural language into richly annotated structures.

2 Motivation

The basic dependency annotation in the Universal Dependencies format introduces labeled edges between tokens in the input string, where each token is a dependent of exactly one other token, with the exception of the root token. While such an annotation layer supports many downstream tasks, there are also phenomena that are hard to capture using single edges between tokens only. The enhanced dependency layer therefore supports a richer level of annotation, where tokens may have more than one parent, and where additional 'empty' tokens may be added to the input string. The enhanced level can be used to account for a range of linguistic phenomena (see Section 3) and to support downstream applications that require representations that capture more aspects of the semantic interpretation of the input.

There are now a number of treebanks that include enhanced dependency annotation. Furthermore, the recent shared tasks on dependency parsing and subsequent work have shown that considerable progress has been made in multilingual dependency parsing. It remains to be seen, however, whether the same is true for enhanced dependency parsing. The challenge is both formal and practical. First, the enhanced representation is a connected graph, possibly containing cycles, while previous work on dependency parsing mostly dealt with rooted trees. Second, as some dependency labels incorporate the lemma of certain dependents and other additional information, the set of labels to be predicted is much larger and language-dependent.

On the other hand, it has been shown that much of the enhanced annotation can be predicted on the basis of the basic UD annotation (Schuster et al., 2017; Nivre et al., 2018). Moreover, most state of the art work in dependency parsing uses a graph-based approach, where the assumption that the output must form a tree is only used in the final step from predicted links to final output. And finally, work on deep-syntax and semantic parsing has shown that accurate mapping of strings into rich graph representations is possible (Oepen et al., 2014, 2015, 2019) and could even lead to state of the art performance for downstream applications as shown by the results of the Extrinsic Evaluation Parsing shared-task (Oepen et al., 2017).

[1] https://iwpt20.sigparse.org

Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task, pages 151–161
Virtual Meeting, July 9, 2020.

3 Enhanced Universal Dependencies

UD version 2[2] states that apart from the morphological and basic dependency annotation layers, strings may be annotated with an additional, enhanced, dependency layer, where the following phenomena can be captured:

- Gapping. To support a linguistically more satisfying treatment of ellipsis, empty tokens can be introduced into the string to represent missing predicates in gapping constructions.
- Coordination. Dependency relations are propagated from the parent of the coordination structure to each conjunct, and from each conjunct to a shared dependent, e.g., a shared subject or object of coordinate verbs.
- Control and raising constructions. The external subject of `xcomp` dependents, if present, can be explicitly marked.
- Relative clauses. The antecedent noun of a relative clause is annotated as a dependent of a node within the relative clause (thus introducing a cycle) and the relative pronoun is annotated as a `ref` dependent of the antecedent noun.
- Case information. Selected dependents (in particular `obl` and `nmod`), if they are marked by morphological case and/or by an adpositional case dependent, can now be labeled as `obl:marker` or `nmod:marker` where `marker` is the lemma of the case dependent and/or the value of the morphological feature `Case`.

All enhancements are optional, so a UD treebank may contain enhanced graphs with one type of enhancement and still lack the other types.

4 Data

The evaluation was done on 17 languages from 4 language families: Arabic, Bulgarian, Czech, Dutch, English, Estonian, Finnish, French, Italian, Latvian, Lithuanian, Polish, Russian, Slovak, Swedish, Tamil, Ukrainian. The language selection is driven simply by the fact that at least partial enhanced representation is available for the given language.

[2] https://universaldependencies.org/u/overview/enhanced-syntax.html

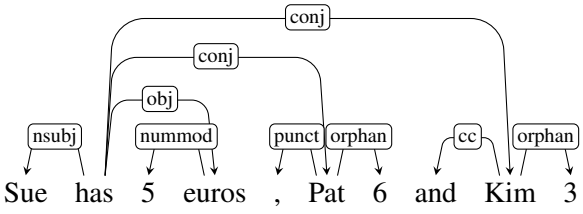

Figure 1: A basic tree of a gapping structure.

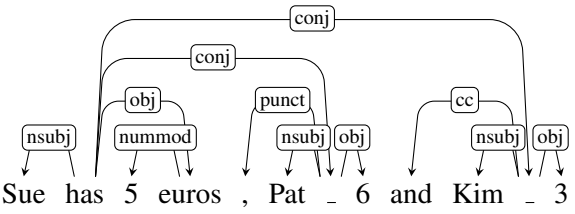

Figure 2: The correct enhanced graph of the gapping structure from Figure 1. "_" are empty nodes.

Training and development data were based on the UD release 2.5 (Zeman et al., 2019) but for several treebanks the enhanced annotation is richer than in UD 2.5. Our goal was to have annotations as uniform and complete as possible. There are only 6 treebanks of 3 languages in UD 2.5 that contain all types of enhancements: Dutch (Alpino and LassySmall), English (EWT and PUD), and Swedish (Talbanken and PUD). For several other languages we obtained new annotations that became part of UD from the next release (2.6) on. For the remaining languages, we applied simple heuristics and added at least some enhancements for the purpose of the shared task, but these annotations are not yet part of the regular UD releases. We only applied our heuristics to the missing enhancement types; we did not attempt to modify the enhancements provided by the data providers. Table 1 gives an overview of enhancements in individual treebanks.

The enhancements differ in how easily and accurately they can be inferred from the basic UD annotation:

- Enhancing relation labels with case information is deterministic. We apply it to the relations `obl`, `nmod`, `advcl` and `acl`. If they have a `case` or `mark` dependent, we add its lowercased lemma (for fixed multiword expressions we glue the lemmas with the "_" character). For `obl` and `nmod` we further examine the `Case` feature and add its lowercased value, if present.

Treebank	UD 2.5	Task	2.6
Arabic PADT	PS	GPS RC	✓
Bulgarian BTB	PSXRC	GPSXRC	
Czech CAC	PS	GPSXRC	✓
Czech FicTree	PS	GPSXRC	✓
Czech PDT	PS	GPSXRC	✓
Czech PUD		GP XRC	✓
Dutch Alpino	GPSXRC	GPSXRC	✓
Dutch LassySmall	GPSXRC	GPSXRC	✓
English EWT	GPSXRC	GPSXRC	✓
English PUD	GPSXRC	GPSXRC	✓
Estonian EDT		GPS RC	(✓)
Estonian EWT	G	GP RC	
Finnish PUD	GP	GP RC	
Finnish TDT	GPSX	GPSXRC	
French FQB		PSX	
French Sequoia		PSX	
Italian ISDT	PSXRC	GPSXRC	
Latvian LVTB	GPSX C	GPSXRC	
Lithuanian ALKS.	PS	GPSXRC	✓
Polish LFG	PSX C	PSXRC	
Polish PDB	PS	GPSXRC	
Polish PUD	PS	GPSXRC	
Russian SynTagRus	G	GP XRC	
Slovak SNK	PS	GPSXRC	✓
Swedish PUD	GPSXRC	GPSXRC	✓
Swedish Talbanken	GPSXRC	GPSXRC	✓
Tamil TTB	PS	PS C	✓
Ukrainian IU	GPSXR	GPSXRC	

Table 1: New annotation for the shared task. Abbreviations: G = gapping; P = parent of coordination; S = shared dependent of coordination; X = external subject of controlled verb; R = relative clause; C = case-enhanced relation label. The check mark in the last column indicates whether the shared task additions also became part of UD 2.6 (only some types for Estonian EDT).

- Linking the parent of coordination to all conjuncts is deterministic.

- Recognizing and transforming relative clauses is easy if relative pronouns can be recognized. This can be tricky in languages where the same pronouns can be used relatively (Figure 3) and interrogatively (Figure 4). We cannot recognize all instances of the latter case reliably; fortunately they do not seem to be too frequent.

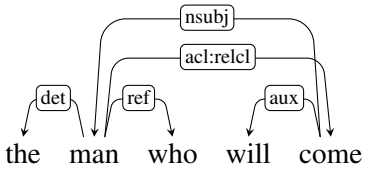

Figure 3: Enhanced graph of a relative clause.

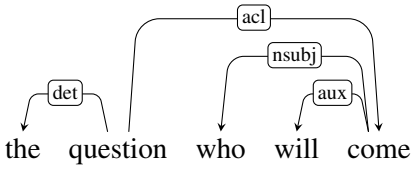

Figure 4: Enhanced graph of an interrogative clause.

- External subjects of xcomp clauses are subjects, objects or oblique dependents of the matrix clause. To find them, we need to know whether the governing verb has subject or object control. We use language-specific verb lists, which can resolve many cases, but not all. If a verb is not on any list, we skip it.

- Gapping can be easily identified by the presence of the orphan relation in the basic tree, insertion of empty nodes is thus trivial. However, we do not know the type of the relation between the empty node and the orphaned dependents. Figure 2 shows a graph where each empty node has one nsubj and one obj dependent. We cannot infer these labels from the basic tree (Figure 1), so we use dep instead.

- Linking conjuncts to shared dependents cannot be done reliably because we cannot know whether a dependent should be shared (this may be sometimes difficult even for a human annotator!) Therefore we do not attempt to add this enhancement to the datasets that do not have it.

Although the UD releases distinguish several different treebanks for some languages, for the purpose of the shared task evaluation we merged all test sets of each language. We wanted to promote robust parsers that are not tightly tied to one particular dataset. Merging treebanks of one language was possible because for almost all languages it holds that treebanks participating in the present task are maintained by the same team, hence no significant treebank-specific annotation decisions are expected. There is one exception, though: Polish. The LFG

Treebank	edeps	% new	% str.n
Arabic PADT	300776	33.88	7.00
Bulgarian BTB	160838	15.30	3.86
Czech CAC	542902	27.61	10.80
Czech FicTree	181370	21.20	9.46
Czech PDT	1612550	24.39	8.20
Czech PUD	20681	26.87	11.42
Dutch Alpino	215595	16.86	4.36
Dutch LassySmall	102130	18.10	4.90
English EWT	267247	17.40	5.17
English PUD	22173	19.58	5.28
Estonian EDT	440974	23.81	1.77
Estonian EWT	29046	26.23	7.52
Finnish PUD	17034	26.27	8.43
Finnish TDT	220061	25.94	9.19
French FQB	24513	2.88	1.55
French Sequoia	73982	6.03	4.70
Italian ISDT	311341	21.39	5.16
Latvian LVTB	238416	23.98	9.56
Lithuanian ALKSNIS	77868	32.25	10.68
Polish LFG	134732	11.17	2.89
Polish PDB	376601	22.82	8.23
Polish PUD	19752	24.61	8.02
Russian SynTagRus	1170014	22.45	6.17
Slovak SNK	111823	20.47	6.12
Swedish PUD	21101	25.25	10.95
Swedish Talbanken	102912	21.19	7.15
Tamil TTB	10408	32.87	7.94
Ukrainian IU	138275	26.48	12.27
total	6945115	23.13	7.09

Table 2: Comparing impact of enhancements in the shared task treebanks where 'edeps' is the number of enhanced dependencies, 'new' is the percentage of edeps that is new when compared to basic UD relations, and 'str.new' are the 'structurally new' dependencies, i.e. dependencies that do not just differ from the basic dependency in having an enhanced dependency label.

treebank uses a different set of relation subtypes than the PDB and PUD treebanks. This is true in the basic trees and it naturally projects to the enhanced graphs. Thus, for example, in LFG the `aux` relation occurs without a subtype (21%), or subtyped `aux:aglt` (65%) or `aux:pass` (14%). In PDB, `aux` occurs without a subtype (21%), or subtyped `aux:clitic` (40%), `aux:cnd` (12%), `aux:imp` (1%) or `aux:pass` (26%). A parser can hardly get the subtypes right when we do not tell it what label dialect is used in the gold data. We can thus expect the labeled attachment score to be less informative in Polish than in the other languages (see Section 6 for alternative evaluation metrics).

Table 2 shows that the effect of enhancements differs quite a bit between the various languages. For instance, the percentage of enhanced dependencies that is 'new', i.e. does not have a corresponding dependency in the basic tree, ranges from 6 to over 30%. Many of these are a consequence of the decision to add the case information to `obl` and other relations, extensions which are relatively easy to capture using a few simple heuristics. Enhanced dependencies that introduce truly novel edges or labels are rarer. The percentage of 'structurally new' relations, i.e. dependencies that differ from the basic dependency in more than just the enhanced label, varies between 2 and 12%.

There are slight differences in how individual languages implement particular enhancement types. Some languages follow earlier proposals for enhanced relation subtypes that are not supported by the current UD guidelines, e.g., external subjects are labeled `nsubj:xsubj`, antecedents of relative clauses are `nsubj:relsubj` or `obj:relobj`, the "case" information is extended to showing conjunction lemma with conjuncts (`conj:and`, `conj:or` etc.) Empty nodes are occasionally used for other ellipsis types than gapping or stripping. A special case is French where diathesis neutralization is encoded in the spirit of Candito et al. (2017).

The data used in the shared task will be permanently available after the shared task at `http://hdl.handle.net/11234/1-3238`.

5 Task

As in the previous dependency parsing shared tasks, participants were expected to go from raw, untokenized, strings to full dependency annotation. The evaluation focused on the enhanced annotation layer, but the participants were encouraged to predict all annotation layers, and the evaluation of the other layers is available on the shared task website.[3] The task was open, in the sense that participants were allowed to use any additional resources they deemed fit (with the exception of UD 2.5 test data) as long as this was announced in advance and the additional resource was freely available to everybody.

[3] `https://universaldependencies.org/iwpt20/`

The submitted system outputs had to be valid CoNLL-U files; if a file was invalid, its score would be zero.[4] The official UD validation script[5] was used to check validity, although only at 'level 2', which means that only basic file format was checked and not the annotation guidelines (e.g., an unknown relation label would not render the file invalid). Still, certain aspects of level-2 validity complicate the prediction of the enhanced graphs, and as the participants were not alerted to individual restrictions beforehand, these restrictions were an unwelcome surprise to them. So the relations can be unknown but can only contain characters from a limited set. The enhanced graph can contain cycles, but not self-loops (a node depending on itself). And most crucially, there must be at least one root node and every node must be reachable via a directed path from at least one root node (rootedness and connectedness). When we saw during the test phase that some teams might not be able to comply with these restrictions, we created a quick-fix script that tries to make the submission valid; however, the solution the script provided for unconnected graphs is not optimal.

In addition to CoNLL-U validity, we also required that systems do not alter any non-whitespace characters when processing the input. This is a pre-requisite for the evaluation, where system-predicted tokens must be aligned with gold-standard tokens; files with modified word forms would be rejected.

6 Evaluation Metrics

The main evaluation metric is ELAS (*labeled attachment score on enhanced dependencies*), where ELAS is defined as F1-score over the set of enhanced dependencies in the system output and the gold standard. Complete edge labels are taken into account, i.e. `obl:on` differs from `obl`. A second metric is EULAS, which differs from ELAS in that only the universal part of the dependency relation label is taken into account. Relation subtypes are ignored, i.e., `obl:on`, `obl:auf`, and `obl` are treated as identical.

As is apparent from Table 1, despite our effort to obtain consistent annotation across all treebanks, there are still treebanks that do not include all enhancements listed in the UD guidelines. Therefore, systems that try to predict all enhancement types for all treebanks might in fact be penalized for predicting more than has been annotated. To give such systems a fair chance, we perform two types of evaluation: 'coarse' and 'qualitative'. In the latter, we ignore dependencies that are specific to enhancement E if the given gold-standard dataset does not include enhancement E. We can trigger individual enhancements on and off separately for each treebank—while the blind input data only distinguishes languages but not treebanks, we still know where each sentence comes from and we can take this information into account during evaluation. The two evaluation methods should give roughly the same result for systems that during training learned to adapt their output to a given treebank, whereas for systems that generally try to predict all possible enhancements, the second method should give more informative results.

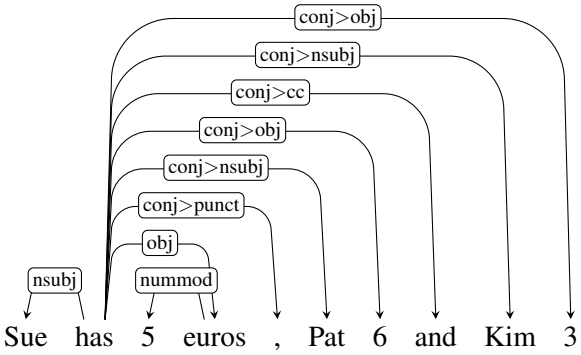

Figure 5: The enhanced graph from Figure 2 after collapsing empty nodes and reflecting the paths in dependency labels.

A final issue we address is the evaluation of empty nodes. A consequence of the treatment of gapping and ellipsis is that some sentences contain additional nodes (numbered 1.1 etc.). It is not guaranteed that gold and system agree on the position in the string where these should appear, but the information encoded by these additional nodes might nevertheless be identical. Thus, such empty nodes should be considered equal even if their string index differs. To ensure that this is the case, we have opted for a solution that basically compiles the information expressed by empty nodes into the dependency label of its dependents. I.e. if a dependent with dependency label `L2` has an empty node `i2.1` as parent which itself is an `L1` dependent of `i1`, its dependency label will be expanded into a path `i1:L1>L2`. This preserves the infor-

[4] https://universaldependencies.org/format.html

[5] https://universaldependencies.org/release_checklist.html#validation

mation that the dependent was an `L2` dependent of 'something' that was itself an `L1` dependent of `i1`, while at the same time removing the potentially conflicting `i2.1` (Figure 5).[6]

7 Approaches

There is quite a bit of variation in the way various teams have addressed the task. For the initial stages of the analysis (tokenization, lemmatization, POS-tagging) some version of UDPipe[7] (Straka et al., 2016), Udify[8] (Kondratyuk and Straka, 2019), and/or Stanza[9] (Qi et al., 2020) is often involved.

Several teams (Orange (Heinecke, 2020), FASTPARSE (Dehouck et al., 2020), UNIPI (Attardi et al., 2020), CLASP (Ek and Bernardy, 2020), ADAPT (Barry et al., 2020)) concentrate on parsing into standard UD, and then add hand-written enhancement rules, sometimes in combination with data-driven heuristics to improve robustness. TurkuNLP (Kanerva et al., 2020) transforms EUD into a representation that is compatible with standard UD by combining multiple edges into a single edge with a complex label, and compiling edges involving empty nodes into complex edge labels (as is done by the evaluation script as well). The total number of edge-labels is reduced by de-lexicalising enhanced edge labels and storing a pointer to the dependent from which the lemma of an enhancement originates in the de-lexicalized edge label. A wide range of parsers (graph-based biaffine, transition-based), and pre-trained embeddings (XLM-R or mBERT or language specific BERTs) is used. Finally, several teams (Emory NLP (He and Choi, 2020), ShanghaiTech (Wang et al., 2020), ADAPT, Køpsala (Hershcovich et al., 2020), RobertNLP (Grünewald and Friedrich, 2020)) do not use conversion (or only to restore de-lexicalized labels), but instead use a graph-based parser that can directly produce enhanced dependency graphs. The output of the graph-based parser is often combined with information from a standard UD parser to ensure well-formedness and connectedness of the resulting graph.

[6] If there are multiple empty nodes in the sentence, we lose the information which orphans were siblings and which were not. Nevertheless, multiple empty nodes in one sentence are extremely rare.

[7] http://ufal.mff.cuni.cz/udpipe

[8] https://github.com/Hyperparticle/udify

[9] https://stanfordnlp.github.io/stanza/

8 Results

We include two baseline results:[10] baseline1 was obtained by taking gold basic UD trees and copying these into the enhanced layer without any modifications. Baseline2 uses UDPipe 1.2 trained on UD 2.5 treebanks[11] and again copies basic UD to the enhanced layer. Both baselines give an impression of how much the enhanced layer differs from the basic layer, where baseline1 makes the unrealistic assumption that parsing into basic UD is perfect.

Table 3 shows that the best three submissions achieve ELAS comparable to LAS for multilingual UD parsing (Zeman et al., 2018; Kondratyuk and Straka, 2019; Kulmizev et al., 2019).

If we compare scores for LAS, EULAS, and ELAS, it can be observed that usually there is a small drop in accuracy when going from LAS to EULAS to ELAS, although the drop from LAS/EULAS to ELAS seems to be larger for some of the systems in the lower half of the table. This suggests that predicting the correct label enhancement is problematic for some approaches.

The EULAS and ELAS scores for the qualitative evaluation (which takes into account differences in the enhancement level of treebanks) are only slightly higher than in the coarse evaluation. It should be noted though, that scores cannot be compared directly, as the coarse evaluation is a macro average over languages, whereas most scores in the qualitative evaluation are macro averages over treebanks. This implies that the data is weighted slightly differently in both averages, which plays a role in the LAS scores being generally a bit higher in the qualitative evaluation. When the qualitative ELAS is averaged over languages (the ELAS-l column in Table 3), the scores become similar to coarse ELAS and no general trend is observable.

Difference between coarse and qualitative evaluation is small. This is due to (a) the fact that this makes a difference for 9 of 28 treebanks only and (b) the fact that some of the phenomena that are ignored in the qualitative evaluation are relatively rare in the data (e.g. ellipsis).

Table 4 shows the best ELAS per language. More detailed results (per language, unofficial re-

[10] We did not include our baseline3 architecture here due to technical issues that prevented us to parse all languages. Encouraging partial results are however available on the shared task website.

[11] Pretrained models (Straka and Straková, 2019) used with default settings, always using the largest available model for the given language. No pretrained word embeddings.

	Coarse			Qualitative			
Team	LAS	EULAS	ELAS	LAS	EULAS	ELAS-t	ELAS-l
baseline1	100.00	96.37	79.86	100.00	96.22	80.70	79.92
baseline2	75.41	72.97	61.07	76.39	73.80	62.32	60.99
TurkuNLP	87.31	85.83	84.50	87.94	86.36	84.63	84.19
Orange	86.79	84.62	82.60	87.78	85.46	83.07	82.52
Emory NLP	86.14	81.26	79.84	87.20	82.34	80.87	79.64
FASTPARSE	77.57	75.96	74.04	78.63	76.99	74.77	73.95
UNIPI	80.74	78.82	72.76	81.61	79.60	73.48	72.82
ShanghaiTech	0.99	73.01	71.74	1.00	73.77	72.40	71.70
CLASP	82.66	80.18	67.85	83.13	80.60	69.20	68.16
ADAPT	84.09	69.42	67.23	84.73	70.10	67.49	67.17
Køpsala	75.41	64.93	62.91	76.39	65.10	62.67	62.72
RobertNLP	5.11	5.26	5.23	6.21	6.39	6.36	5.24

Table 3: Evaluation results on the test data. LAS is the evaluation of the basic tree, EULAS and ELAS evaluate the enhanced graph. In Coarse, the score is the macro average over languages, in Qualitative, the score for LAS and EULAS is the macro average over treebanks. ELAS-t gives the macro average over treebanks, and ELAS-l the macro average over languages. RobertNLP submitted only the English data.

Language	**Team**	**ELAS**
Arabic	TurkuNLP	77.82
Bulgarian	TurkuNLP	90.73
Czech	TurkuNLP	87.51
Dutch	Orange	85.14
English	RobertNLP	88.94
Estonian	TurkuNLP	84.54
Finnish	TurkuNLP	89.49
French	Emory NLP	86.23
Italian	TurkuNLP	91.54
Latvian	TurkuNLP	84.94
Lithuanian	TurkuNLP	77.64
Polish	TurkuNLP	84.64
Russian	TurkuNLP	90.69
Slovak	TurkuNLP	88.56
Swedish	TurkuNLP	85.64
Tamil	Orange	64.23
Ukrainian	TurkuNLP	87.22

Table 4: Best results per language (Coarse).

sults) are available on the results page of the shared task website.[12]

9 Post Shared Task Unofficial Results

A number of teams have submitted runs on the test data after the deadline for the official evaluation, an overview in given in Table 5. In some cases, these are runs that fix validation issues and that result in considerably higher scores (i.e., ShanghaiTech). In other cases, these unofficial runs are experiments with various components of the system architecture. The reader should consult the system description papers for further discussion of these results.

10 Conclusions

This shared task was the first attempt at a coordinated evaluation effort on parsing enhanced universal dependencies. While a large part of the methodology could be adopted from the previous CoNLL shared tasks on parsing into UD, a number of issues did require attention.

First, providing training and test data is complicated by the fact that not all treebanks in the UD repository include the same level of enhancements. This makes training a single, multilingual, model, harder than it ought to be, as annotation style differs per treebank. For evaluation, different enhancement levels pose a problem as it is unclear to what extent 'overannotating' data should be considered an error. As Table 1 illustrates, the situation has improved already considerably for UD release 2.6.

Another issue for validation is the status of 'empty' nodes. The position in the string of such nodes is not defined by the guidelines, and therefore one may expect mismatches between gold and system data. Our solution to this issue is described in Section 6. For future tasks, however, it might

[12] https://universaldependencies.org/iwpt20/Results.html

	Coarse			Qualitative			
Team	LAS	EULAS	ELAS	LAS	EULAS	ELAS-t	ELAS-l
ShanghaiTech	1.05	86.54	85.06	1.04	87.23	85.63	84.96
ADAPT	84.91	82.25	79.95	85.60	83.12	80.15	79.89
FASTPARSE	79.85	78.27	76.48	80.82	79.20	77.13	76.36
Køpsala	75.41	78.92	76.48	76.39	79.28	76.33	76.28
UNIPI	84.32	82.32	75.92	85.76	83.60	77.16	75.92

Table 5: Post Shared Task evaluation results on the test data.

be worthwhile to investigate whether a different representation of such nodes in the data files or an alternative evaluation strategy is needed.

Several systems struggled with the validation requirements of enhanced UD. While an enhanced graph may contain nodes with more than one parent, may contain cycles, and may have multiple root nodes, there are still constraints that an enhanced UD graph must comply with, such as that the graph must be connected and that there should be one or more 'root' nodes from which all other nodes are reachable. In future tasks, the restrictions should be more carefully described in advance.

The results of the shared task illustrate that there is quite a wide variety in the way that the problem of parsing into enhanced universal dependencies can be approached, with some systems sticking closer to traditional approaches for parsing UD, and dealing with the enhancements in a conversion script, while other systems output a graph directly. The scores indicate that while parsing into enhanced UD is harder than parsing into UD, the drop in performance is minimal for most systems, which suggests that the challenges posed by the annotation format of enhanced UD are not an obstacle for accurate parsing.

Acknowledgments

We heartily thank everyone involved in the development of the Enhanced UD treebanks and who made this shared task possible.
This work has been partially supported by the LUSyD project, grant 20-16819X of the Czech Science Foundation (GAČR). The second author was partly funded by two French National Research Agency projects, PARSITI (ANR-16-CE33-0021) and SoSweet (ANR-15-CE38-0011).

References

Giuseppe Attardi, Daniele Sartiano, and Maria Simi. 2020. Linear neural parsing and hybrid enhancement for Enhanced Universal Dependencies. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

James Barry, Joachim Wagner, and Jennifer Foster. 2020. The ADAPT Enhanced Dependency Parser at the IWPT 2020 Shared Task. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

Marie Candito, Bruno Guillaume, Guy Perrier, and Djamé Seddah. 2017. Enhanced UD dependencies with neutralized diathesis alternation. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 42–53, Pisa, Italy.

Mathieu Dehouck, Mark Anderson, and Carlos Gómez-Rodríguez. 2020. Efficient EUD parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

Adam Ek and Jean-Philippe Bernardy. 2020. How much of enhanced UD is contained in UD? In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

Stefan Grünewald and Annemarie Friedrich. 2020. Robertnlp at the IWPT 2020 Shared Task: Surprisingly Simple Enhanced UD Parsing for English. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

Han He and Jinho D. Choi. 2020. Adaptation of Multilingual Transformer Encoder for Robust Enhanced

Universal Dependency Parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

Johannes Heinecke. 2020. Hybrid Enhanced Universal Dependencies Parsing. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

Daniel Hershcovich, Miryam de Lhoneux, Artur Kulmizev, Elham Pejhan, and Joakim Nivre. 2020. Køpsala: Transition-Based Graph Parsing via Efficient Training and Effective Encoding. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

Jenna Kanerva, Filip Ginter, and Sampo Pyysalo. 2020. Turku Enhanced Parser Pipeline: From Raw Text to Enhanced Graphs in the IWPT 2020 Shared Task. In *Proceedings of the 16th International Conference on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume)*. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795.

Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2755–2768, Hong Kong, China. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, pages 4027–4036, Paris, France. European Language Resources Association.

Joakim Nivre, Paola Marongiu, Filip Ginter, Jenna Kanerva, Simonetta Montemagni, Sebastian Schuster, and Maria Simi. 2018. Enhancing universal dependency treebanks: A case study. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 102–107.

Stephan Oepen, Omri Abend, Jan Hajič, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdeňka Urešová. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. Association for Computational Linguistics.

Stephan Oepen, Jari Björne, Richard Johansson, Emanuele Lapponi, Filip Ginter, Erik Velldal, and Lilja Øvrelid. 2017. The 2017 Shared Task on Extrinsic Parser Evaluation (EPE 2017).

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *In Association for Computational Linguistics (ACL) System Demonstrations*, Seattle, WA, USA.

Sebastian Schuster, Eric De La Clergerie, Marie Candito, Benoît Sagot, Christopher D. Manning, and Djamé Seddah. 2017. Paris and Stanford at EPE 2017: Downstream evaluation of graph-based dependency representations.

Milan Straka, Jan Hajič, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Milan Straka and Jana Straková. 2019. Universal dependencies 2.5 models for UDPipe (2019-12-06). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Xinyu Wang, Yong Jiang, and Kewei Tu. 2020. Enhanced Universal Dependency Parsing with Second-Order Inference and Mixture of Training Data. In *Proceedings of the 16th International Conference*

on Parsing Technologies and the IWPT 2020 Shared Task on Parsing into Enhanced Universal Dependencies (this volume). Association for Computational Linguistics.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Sandra Bellato, Kepa Bengoetxea, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Savas Cetin, Fabricio Chalub, Jinho Choi, Yongseok Cho, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Richárd Farkas, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Kim Gerdes, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Nizar Habash, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Kim Harris, Dag Haug, Johannes Heinecke, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Jena Hwang, Takumi Ikeda, Radu Ion, Elena Irimia, Ọlájídé Ishola, Tomáš Jelínek, Anders Johannsen, Fredrik Jørgensen, Markus Juutinen, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Kamil Kopacewicz, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Sookyoung Kwak, Veronika Laippala, Lorenzo Lambertino, Lucia Lam, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phng Lê H`ông, Alessandro Lenci, Saran Lertpradit, Herman Leung, Cheuk Ying Li, Josie Li, Keying Li, KyungTae Lim, Maria Liovina, Yuan Li, Nikola Ljubešić, Olga Loginova, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, Simonetta Montemagni, Amir More, Laura Moreno Romero, Keiko Sophie Mori, Tomohiko Morioka, Shinsuke Mori, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Lng Nguy˜ên Thị, Huy`ên Nguy˜ên Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Petya Osenova, Robert Östling, Lilja Øvrelid, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Guy Perrier, Daria Petrova, Slav Petrov, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Loganathan Ramasamy, Taraka Rama, Carlos Ramisch, Vinit Ravishankar, Livy Real, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Mykhailo Romanenko, Rudolf Rosa, Davide Rovati, Valentin Roșca, Olga Rudina, Jack Rueter, Shoval Sadde, Benoît Sagot, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Nathan Schneider, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Muh Shohibussirri, Dmitry Sichinava, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Antonio Stella, Milan Straka, Jana Strnadová, Alane Suhr, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Takaaki Tanaka, Isabelle Tellier, Guillaume Thomas, Liisi Torga, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Daniel van Niekerk, Gert-

jan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Zhuoran Yu, Zdeněk Žabokrtský, Amir Zeldes, Manying Zhang, and Hanzhi Zhu. 2019. Universal dependencies 2.5. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gökırmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.