An Error Analysis Framework for Shallow Surface Realisation

Anastasia Shimorina, Yannick Parmentier, Claire Gardent

An Error Analysis Framework for Shallow Surface Realization

Anastasia Shimorina Yannick Parmentier Claire Gardent

Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France
{anastasia.shimorina, yannick.parmentier, claire.gardent}@loria.fr

Abstract

The metrics standardly used to evaluate Natural Language Generation (NLG) models, such as BLEU or METEOR, fail to provide information on which linguistic factors impact performance. Focusing on Surface Realization (SR), the task of converting an unordered dependency tree into a well-formed sentence, we propose a framework for error analysis which permits identifying which features of the input affect the models' results. This framework consists of two main components: (i) correlation analyses between a wide range of syntactic metrics and standard performance metrics and (ii) a set of techniques to automatically identify syntactic constructs that often co-occur with low performance scores. We demonstrate the advantages of our framework by performing error analysis on the results of 174 system runs submitted to the Multilingual SR shared tasks; we show that dependency edge accuracy correlate with automatic metrics thereby providing a more interpretable basis for evaluation; and we suggest ways in which our framework could be used to improve models and data. The framework is available in the form of a toolkit which can be used both by campaign organizers to provide detailed, linguistically interpretable feedback on the state of the art in multilingual SR, and by individual researchers to improve models and datasets.[1]

1 Introduction

Surface Realization (SR) is a natural language generation task that consists in converting a linguistic representation into a well-formed sentence.

SR is a key module in pipeline generation models, where it is usually the last item in a pipeline of modules designed to convert the input (knowledge graph, tabular data, numerical data) into a text. While end-to-end generation models have been proposed that do away with such pipeline architecture and therefore with SR, pipeline generation models (Dušek and Jurčíček, 2016; Castro Ferreira et al., 2019; Elder et al., 2019; Moryossef et al., 2019) have been shown to perform on a par with these end-to-end models while providing increased controllability and interpretability (each step of the pipeline provides explicit intermediate representations that can be examined and evaluated).

As illustrated in, for example, Dušek and Jurčíček (2016), Elder et al. (2019), and Li (2015), SR also has potential applications in tasks such as summarization and dialogue response generation. In such approaches, shallow dependency trees are viewed as intermediate structures used to mediate between input and output, and SR permits regenerating a summary or a dialogue turn from these intermediate structures.

Finally, multilingual SR is an important task in its own right in that it permits a detailed evaluation of how neural models handle the varying word order and morphology of the different natural languages. While neural language models are powerful at producing high quality text, the results of the multilingual SR tasks (Mille et al., 2018, 2019) clearly show that the generation, from shallow dependency trees, of morphologically and syntactically correct sentences in multiple languages remains an open problem.

As the use of multiple input formats made the comparison and evaluation of existing surface realisers difficult, Belz et al. (2011) and Mille et al. (2018, 2019) organized the SR shared tasks, which provide two standardized input formats for surface realizers: deep and shallow dependency trees. Shallow dependency trees are unordered, lemmatized dependency trees. Deep dependency trees include semantic rather than syntactic relations and abstract over function words.

While the SR tasks provide a common benchmark on which to evaluate and compare SR systems, the evaluation protocol they use (automatic

[1] Our code and settings to reproduce the experiments are available at https://gitlab.com/shimorina/tacl-2021.

Transactions of the Association for Computational Linguistics, vol. 9, pp. 429–446, 2021. https://doi.org/10.1162/tacl_a_00376
Action Editor: Roi Reichart. Submission batch: 11/2020; Revision batch: 1/2020; Published 4/2021.

metrics and human evaluation) does not support a detailed error analysis. Metrics (BLEU, DIST, NIST, METEOR, TER) and human assessments are reported on the system level, and so do not provide a detailed feedback for each participant. Neither do they give information about which syntactic phenomena impact performance.

In this work, we propose a framework for error analysis that allows for an interpretable, linguistically informed analysis of SR results. While shallow surface realization involves both determining word order (linearization) and inflecting lemmas (morphological realization), since inflection error detection is already covered in morphological shared tasks (Cotterell et al., 2017; Gorman et al., 2019), we focus on error analysis for word order.

Motivated by extensive linguistic studies that deal with syntactic dependencies and their relation to cognitive language processing (Liu, 2008; Futrell et al., 2015; Kahane et al., 2017), we investigate word ordering performance in SR models given various tree-based metrics. Specifically, we explore the hypothesis according to which these metrics, which provide a measure of the SR input complexity, correlate with automatic metrics commonly used in NLG. We find that Dependency Edge Accuracy (DEA) correlates with BLEU, which suggests that DEA could be used as an alternative, more interpretable, automatic evaluation metric for surface realizers.

We apply our framework to the results of two evaluation campaigns and demonstrate how it can be used to highlight some global results about the state of the art (e.g., that certain dependency relations such as the *list* dependency have low accuracy across the board for all 174 submitted runs).

We indicate various ways in which our error analysis framework could be used to improve a model or a dataset, thereby arguing for approaches to model and dataset improvement that are more linguistically guided.

Finally, we make our code available in the form of a toolkit that can be used both by campaign organizers to provide a detailed feedback on the state of the art for surface realization and by researchers to better analyze, interpret, and improve their models.

2 Related Work

There has been a long tradition in NLP exploring syntactic and semantic evaluation measures based on linguistic structures (Liu and Gildea, 2005; Mehay and Brew, 2007; Giménez and Màrquez, 2009; Tratz and Hovy, 2009; Lo et al., 2012). In particular, dependency-based automatic metrics have been developed for summarization (Hovy et al., 2005; Katragadda, 2009; Owczarzak, 2009) and machine translation (Owczarzak et al., 2007; Yu et al., 2014). Relations between metrics were also studied: Dang and Owczarzak (2008) found that automatic metrics perform on a par with the dependency-based metric of Hovy et al. (2005) while evaluating summaries. The closest research to ours, which focused on evaluating how dependency-based metrics correlate with human ratings, is Cahill (2009), who showed that syntactic-based metrics perform equally well as compared to automatic metrics in terms of their correlation with human judgments for a German surface realizer.

Researchers, working on SR and word ordering, have been resorting to different metrics to report their models' performance. Zhang et al. (2012), Zhang (2013), Zhang and Clark (2015), Puduppully et al. (2016), and Song et al. (2018) used BLEU; Schmaltz et al. (2016) parsed their outputs and calculated the UAS parsing metric; Filippova and Strube (2009) used Kendall correlation together with edit-distance to account for English word order. Similarly, Dyer (2019) used Spearman correlation between produced and gold word order for a dozen of languages. White and Rajkumar (2012), in their CCG-based realization, calculated average dependency lengths between grammar-generated sentences and gold standard. Gardent and Narayan (2012) and Narayan and Gardent (2012) proposed an error mining algorithm for generation grammars to identify the most likely sources of failures, when generating from dependency trees. Their algorithm mines suspicious subtrees in a dependency tree, which are likely to cause errors. King and White (2018) drew attention to their model performance for non-projective sentences. Puzikov et al. (2019) assessed their binary classifier for word ordering using the accuracy of predicting the position of a dependent with respect to its head, and a sibling. Yu et al. (2019) showed that, for their system, error rates correlate with word order freedom, and reported linearization error rates for some frequent dependency types. In a similar vein, Shimorina and Gardent (2019) looked at their system performance in terms of dependency relations,

which shed light on the differences between their non-delexicalized and delexicalized models.

In sum, multiple metrics and tools have been developed by individual researchers to evaluate and interpret their model results: dependency-based metrics, correlation between these metrics and human ratings, performance on projective vs. non-projective input, linearization error rate, and so forth. At a more global level, however, automatic metrics and human evaluation continue to be massively used.

In this study, we gather a set of linguistically informed, interpretable metrics and tools within a unified framework, apply this framework to the results of two evaluation campaigns (174 participant submissions) and generally argue for a more interpretable evaluation approach for surface realizers.

3 Framework for Error Analysis

Our error analysis framework gathers a set of performance metrics together with a wide range of tree-based metrics designed to measure the syntactic complexity of the sentence to be generated. We apply correlation tests between these two types of metrics and mine a model output to automatically identify the syntactic constructs that often co-occur with low performance scores.

3.1 Syntactic Complexity Metrics

To measure syntactic complexity, we use several metrics commonly used for dependency trees (**tree depth and length, mean dependency distance**) as well as the ratio, in a test set, of sentences with **non-projective structures**.

We also consider the entropy of the dependency relations and a set of metrics based on ''flux'' recently proposed by Kahane et al. (2017).

Flux. The flux is defined for each inter-word position (e.g., 5–6 in Figure 1). Given the inter-word position (i, j), the flux of (i, j) is the set of edges (d, k, l) such that d is a dependency relation, $k \leq i$ and $j \leq l$. For example, in Figure 1 the flux for the inter-word position between the nodes 5 and 6 is {*(nmod, 4, 8), (case, 5, 8)*} and {*(nmod, 4, 8), (case, 5, 8), (compound, 6, 8), (compound, 7, 8)*} for the position between the nodes 7 and 8.

The **flux size** is its cardinality, that is, the number of edges it contains: 2 for 5–6 and 4 for 7–8.

The **flux weight** is the size of the largest disjoint subset of edges in the flux (Kahane et al., 2017, p. 74). A set of edges is disjoint if the edges it contains do not share any node. For instance, in the inter-word position 5–6, *nmod* and *case* share a common node 8, so the flux weight is 1 (i.e., it was impossible to find two disjoint edges). The idea behind the flux-based metrics was to try accounting for cognitive complexity of syntactic structures, in the same fashion as in Miller (1956), who showed a processing limitation of syntactic constituents in a spoken language.

For each reference dependency tree, we calculate the metrics listed in Table 1. These can then be averaged over different dimensions (all runs, all runs of a given participant, runs on a given corpus, language, etc.). Table 2 shows the statistics obtained for each corpus used in the SR shared tasks. We refer the reader to the Universal Dependencies project[2] to learn more about differences between specific treebanks.

Dependency Relation Entropy. Entropy has been used in typological studies to quantify word order freedom across languages (Liu, 2010; Futrell et al., 2015; Gulordava and Merlo, 2016). It gives an estimate of how regular or irregular a dependency relation is with respect to word order. A relation d with high entropy indicates that d-dependents sometimes occur to the left and sometimes to the right of their head—that is, their order is not fixed.

The entropy H of a dependency relation d is calculated as

$$H(d) = -p(L) \times \log_2(p(L)) - p(R) \times \log_2(p(R))$$

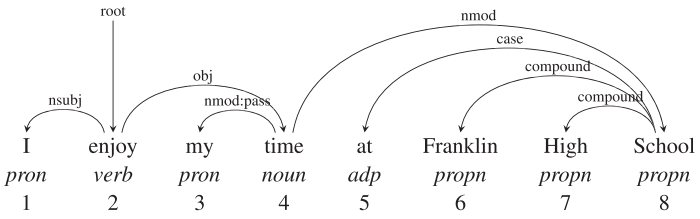
where $p(L)$ is the probability for a dependent to be on the left from the head, and $p(R)$ is the probability for a dependent to be on the right from the head. For instance, if the dependency relation *amod* is found to be head-final 20 times in a treebank, and head-initial 80 times, its entropy is equal to 0.72. Entropy ranges from 0 to 1: Values close to zero indicate low word order freedom; values close to one mark high variation in head directionality.

3.2 Performance Metrics

Performance is assessed using sentence-level BLEU-4, DEA, and human evaluation scores.

DEA. DEA measures how many edges from a reference tree can be found in a system output, given the gold lemmas and dependency distance

[2] https://universaldependencies.org/.



System output (lemmatized): *I enjoy my time at High Franklin School*
BLEU= 0.65; dep edge accuracy = 0.71

System output (final): *I enjoyed my time at High Franklin School*
fluency= 1; adequacy = 0.7

Figure 1: A reference UD dependency tree (nodes are lemmas) and a possible SR model output. The final output is used to compute human judgments and the lemmatized output to compute BLEU and dependency edge accuracy (both are given without punctuation).

Syntactic Complexity	Explanation
tree depth	the depth of the deepest node $\{3\}$
tree length	number of nodes $\{8\}$
mean dependency distance	average distance between a head and a dependent. For a dependency linking two adjacent nodes, the distance is equal to one (e.g., *nsubj* in Figure 1). $\{(1+2+1+4+1+2+3)/7 = 2\}$
mean flux size	average flux size of each inter-word position $\{(1+1+2+1+2+3+4)/7 = 2\}$
mean flux weight	average flux weight of each inter-word position $\{(1+1+1+1+1+1+1)/7 = 1\}$
mean arity	average number of direct dependents of a node $\{(0+2+0+2+0+0+0+3)/8 = 0.875\}$
projectivity	True if the sentence has a projective parse tree (there are no crossing dependency edges and/or projection lines) $\{$*True*$\}$

Table 1: Metrics for Syntactic Complexity of a sentence (the values in braces indicate the corresponding value for the tree in Figure 1).

as markers. An edge is represented as a triple (head lemma, dependent lemma, distance), for example, (I, enjoy, -1) or (time, school, +4) in Figure 1.[3] In the output, the same triples can be found based on the lemmas, the direction (after or before the head), and the dependency distance. In our example, two out of the seven dependency relations cannot be found in the output: (school, high, -1) and (school, franklin, -2). Thus, DEA is 0.71 $(5/7)$.

Human Evaluation Scores. The framework include sentence-level z-scores for Adequacy and Fluency[4] reported in the SR'18 and SR'19 shared tasks. The z-scores were calculated on the set of all raw scores by the given annotator using each annotator's mean and standard deviation. Note that those were available for a sample of test instances for some languages only and were calculated using the final system outputs, rather than the lemmatized ones.

3.3 Correlation Tests

The majority of our metrics are numerical, which allows us to measure dependence between them using correlation. One of the metrics—projectivity—is nominal, so we apply a non-parametric test to measure whether two independent samples (''projective sentences'' and ''non-projective sentences'') have the same distribution of scores.

[3]We report signed values for dependency distance, rather than absolute ones, to account for the dependent position—after or before the head.

[4]In the original papers called *Meaning Similarity* and *Readability*, respectively (Mille et al., 2018, 2019).

3.4 Error Mining

Tree error mining of Narayan and Gardent (2012) was initially developed to explain errors in grammar-based generators. The algorithm takes as input two groups of dependency trees: Those whose derivation was covered (P for Pass) and those whose derivation was not covered (F for Fail) by the generator. Based on these two groups, the algorithm computes a suspicion score S for each subtree f in the input data as follows:

$$S(f) = \frac{1}{2}\left(\frac{c(f|F)}{c(f)}\ln c(f) + \frac{c(\neg f|P)}{c(f)}\ln c(\neg f)\right)$$

$c(f)$ is the number of sentences containing a subtree f, $c(\neg f)$ is the number of sentences where f is not present, $c(f|F)$ is the number of sentences containing f for which generation failed, and $c(\neg f|P)$ is the number of sentences not containing f for which generation succeeded. Intuitively, a high suspicion score indicates a subtree (a syntactic construct) in the input data which often co-occurs with failure and seldom with success. The score is inspired from the decision tree classifier information gain metrics (Quinlan, 1986), which is there used to cluster the input data into subclusters with maximal purity and adapted to take into account the degree to which a subtree associates with failure rather than the entropy of the subclusters.

To imitate those two groups of successful and unsuccessful generation, we adapted a threshold based on BLEU. All the instances in a model output are divided into two parts: The first quartile (25% of instances)[5] with a low sentence-level BLEU was considered as failure, the rest—as success. Error mining can then be used to automatically identify subtrees of the input tree that often co-occur with failure and rarely with success. Moreover, mining can be applied to trees decorated with any combination of lemmas, dependency relations and/or POS tags.

4 Data and Experimental Setting

We apply our error analysis methods to 174 system outputs (runs) submitted to the shallow track of SR'18 and SR'19 shared tasks (Mille et al., 2018, 2019). For each generated sentence in the submissions, we compute the metrics described in the preceding section as follows.

[5]It is our empirical choice. Any other threshold can also be chosen.

Computing Syntactic Complexity Metrics. Tree-based metrics, dependency relation entropy and projectivity are computed on the gold parse trees from Universal Dependencies v2.0 and v2.3 (Nivre et al., 2017) for SR'18 and SR'19, respectively. Following common practice in dependency linguistics computational studies, punctuation marks were stripped from the reference trees (based on *punct* dependency relation). If a node to be removed had children, these were assigned to the parent of the node.

Computing Performance Metrics. We compute sentence-level BLEU-4 with the smoothing method 2 from Chen and Cherry (2014), implemented in NLTK.[6]

To compute dependency edge accuracy, we process systems' outputs to allow for comparison with the lemmatized dependency tree of the reference sentence. Systems' outputs were tokenized and lemmatized; contractions were also split to match lemmas in the UD treebanks. Finally, to be consistent with punctuation-less references, punctuation was also removed from systems' outputs. The preprocessing was done with the stanfordnlp library (Qi et al., 2018).

For human judgments, we collect those provided by the shared tasks for a sample of test data and for some languages (*en, es, fr* for SR'18 and *es_ancora, en_ewt, ru_syntagrus, zh_gsd* for SR'19). Table 2 shows how many submissions each language received.

Computing Correlation. For all numerical variables, we assess the relationship between rankings of two variables using Spearman's ρ correlation. When calculating correlation coefficients, missing values were ignored (that was the case for human evaluations). Correlations were calculated separately for each submission (one system run for one corpus). Because up to 45 comparisons can be made for one submission, we controlled for the multiple testing problem using the Holm-Bonferroni method while doing a significance test. We also calculated means and medians of the correlations for each corpus (all submissions mixed), for each team (a team has multiple submissions), and average correlations through all the 174 submissions.

[6]We do not include other automatic n-gram-based metrics used in the SR shared tasks because they usually correlate with each other.

		S	count	depth	length	MDD	MFS	MFW	MA	NP
SR'18	ar (padt)	3	676	7.37±3.29	38.5±30.38	2.61±0.93	2.61±0.93	1.44±0.26	0.94±0.08	1.48
	cs (pdt)	2	9,876	3.95±1.99	14.49±9.43	2.12±0.74	2.12±0.74	1.19±0.29	0.86±0.18	9.91
	es (ancora)	6	1,719	5.21±2.2	26.88±15.7	2.47±0.66	2.47±0.66	1.33±0.25	0.93±0.09	2.39
	en (ewt)	8	2,061	2.71±1.88	10.57±9.55	1.86±0.95	1.86±0.95	1.02±0.42	0.75±0.3	1.65
	fi (tdt)	3	1,525	3.48±1.81	11.42±7.22	2.02±0.62	2.02±0.62	1.16±0.23	0.86±0.12	5.57
	fr (gsd)	5	416	4.33±1.75	21.21±12.57	2.44±0.59	2.44±0.59	1.28±0.25	0.93±0.07	2.16
	it (isdt)	4	480	4.38±2.23	19.14±14.07	2.19±0.61	2.19±0.61	1.23±0.23	0.91±0.06	2.29
	nl (alpino)	4	685	3.74±1.86	15.03±9.11	2.48±1.05	2.48±1.05	1.21±0.39	0.85±0.22	20.15
	pt (bosque)	4	476	4.32±2.12	18.58±12.11	2.25±0.63	2.25±0.63	1.23±0.27	0.9±0.13	4.20
	ru (syntagrus)	2	6,366	4.1±1.96	14.65±9.14	2.12±0.66	2.12±0.66	1.23±0.27	0.88±0.13	8.37
SR'19	ar_padt	4	680	7.38±3.28	38.54±30.34	2.6±0.93	2.6±0.93	1.45±0.26	0.94±0.08	1.76
	en_ewt	5	2,077	2.72±1.88	10.6±9.62	1.87±0.95	1.87±0.95	1.02±0.42	0.75±0.3	1.54
	en_gum	11	778	3.69±1.91	15.0±10.63	2.14±0.75	2.14±0.75	1.16±0.31	0.85±0.2	3.08
	en_lines	11	914	3.55±1.6	14.97±9.56	2.27±0.62	2.27±0.62	1.2±0.23	0.89±0.11	4.60
	en_partut	11	153	4.52±2.01	20.06±9.77	2.48±0.51	2.48±0.51	1.26±0.21	0.93±0.05	0.65
	es_ancora	6	1,721	5.2±2.2	26.87±15.7	2.47±0.66	2.47±0.66	1.33±0.25	0.93±0.09	2.38
	es_gsd	6	426	5.06±2.25	25.18±16.43	2.41±0.57	2.41±0.57	1.31±0.23	0.94±0.05	4.69
	fr_gsd	7	416	4.41±1.78	21.22±12.58	2.41±0.58	2.41±0.58	1.28±0.25	0.93±0.07	1.20
	fr_partut	7	110	4.85±1.82	21.84±10.01	2.44±0.46	2.44±0.46	1.29±0.21	0.94±0.03	0.91
	fr_sequoia	7	456	4.01±2.21	19.66±15.61	2.13±0.84	2.13±0.84	1.16±0.37	0.84±0.25	0.88
	hi_hdtb	5	1,684	4.19±1.48	19.6±8.99	2.96±0.82	2.96±0.82	1.48±0.23	0.94±0.03	8.91
	id_gsd	5	557	4.57±1.85	18.02±12.39	2.04±0.54	2.04±0.54	1.22±0.2	0.92±0.07	0.72
	ja_gsd	6	551	4.36±1.97	20.25±13.35	2.43±0.66	2.43±0.66	1.4±0.32	0.92±0.09	0.00
	ko_gsd	5	989	3.59±1.78	10.29±6.77	2.21±0.79	2.21±0.79	1.33±0.36	0.86±0.1	9.20
	ko_kaist	4	2,287	3.86±1.54	11.0±4.56	2.27±0.67	2.27±0.67	1.44±0.32	0.89±0.07	19.15
	pt_bosque	5	477	4.32±2.11	18.57±12.09	2.25±0.63	2.25±0.63	1.23±0.27	0.9±0.13	4.40
	pt_gsd	5	1,204	4.85±1.87	22.74±12.2	2.39±0.55	2.39±0.55	1.31±0.23	0.94±0.05	1.66
	ru_gsd	5	601	4.11±1.69	15.83±10.24	2.12±0.69	2.12±0.69	1.24±0.21	0.91±0.06	4.49
	ru_syntagrus	4	6,491	4.08±1.94	14.78±9.24	2.13±0.65	2.13±0.65	1.23±0.26	0.88±0.13	6.49
	zh_gsd	7	500	4.22±1.08	20.64±10.17	2.98±0.84	2.98±0.84	1.46±0.27	0.94±0.03	0.40

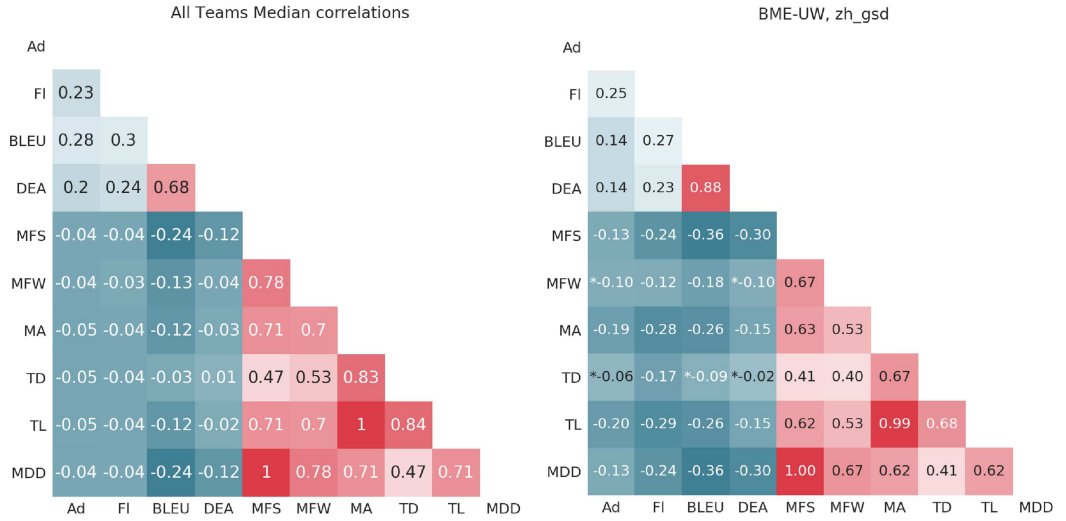
Table 2: Descriptive statistics (mean and stdev apart from the first two and the last column) for the UD treebanks used in SR'18 (UD v2.0) and SR'19 (UD v2.3). S: number of submissions, count: number of sentences in a test set, MDD: mean dependency distance, MFS: mean flux size, MFW: mean flux weight, MA: mean arity, NP: percentage of non-projective sentences. For the tree-based metrics (MDD, MFS, MFW, MA), macro-average values are reported. For SR'18, we follow the notation for treebanks as used in the shared task (only language code); in parentheses we list treebank names.

For projectivity (nominal variable) we use a Mann–Whitney U test to determine whether there is a difference in performance between projective and non-projective sentences. We ran three tests where performance was defined in terms of BLEU, fluency, and adequacy. As for some corpora, the count of non-projective sentences in their test set is low (e.g., 1.56% in *en_ewt*), we ran the test on the corpora that have more than 5% of non-projective sentences, that is, *cs* (10%), *fi* (6%), *nl* (20%), and *ru* (8%) for SR'18, and *hi_hdtb* (9%), *ko_gsd* (9%), *ko_kaist* (19%), and *ru_syntagrus* (6%) for SR'19. For the calculation of the Mann–Whitney U test, we used scipy-1.4.1. Similar to the correlation analysis, the test was calculated separately for each submission and for each corpus.

Mining the Input Trees. The error mining algorithm was run for each submission separately and with three different settings: (*i*) dependency relations (dep); *(ii)* POS tags (POS); *(iii)* dependency relations and POS tags (POS-dep).

5 Error Analysis

We analyze results focusing successively on: tree-based syntactic complexity (are sentences with



(a) Median Spearman ρ coefficients between metrics (all submissions considered).

(b) Spearman ρ coefficients between metrics for BME-UW for zh_gsd.

Figure 2: Spearman ρ coefficients between metrics. Ad: adequacy z-score, Fl: fluency z-score, DEA: dependency edge accuracy, MFS: mean flux size, MFW: mean flux weight, MA: mean arity, TD: tree depth, TL: tree length, MDD: mean dependency distance. * – non-significant coefficients at $\alpha = 0.05$ corrected with the Holm-Bonferroni method for multiple hypotheses testing.

more complex syntactic trees harder to generate?), projectivity (how much does non-projectivity impact results?), entropy (how much do word order variations affect performance?), DEA and error mining (which syntactic constructions lead to decreased scores?).

5.1 Tree-Based Syntactic Complexity

We examine correlation tests results for all metrics on the system level (all submissions together) and for a single model, the BME-UW system (Kovács et al., 2019) on a single corpus/language (*zh_gsd*, Chinese). Figure 2a shows median Spearman ρ coefficients across all the 174 submissions, and Figure 2b shows the coefficients for the BME-UW system on the *zh_gsd* corpus.

We investigate both correlations between syntactic complexity and performance metrics and within each category. Similar observations can be made for both settings.

Correlation between Performance Metrics. As often remarked in the NLG context (Stent et al., 2005; Novikova et al., 2017; Reiter, 2018), BLEU shows a weak correlation with Fluency and Adequacy on the sentence level. Similarly, dependency edge accuracy shows weak correlations with human judgments ($\rho_{ad} = 0.2$ and $\rho_{fl} = 0.24$ for the median; $\rho_{ad} = 0.14$ and $\rho_{fl} = 0.23$ for BME-UW).[7]

In contrast, BLEU shows a strong correlation with dependency edge accuracy (median: $\rho = 0.68$; BME-UW: $\rho = 0.88$). Contrary to BLEU however, DEA has a direct linguistic interpretation (it indicates which dependency relations are harder to handle) and can be exploited to analyze and improve a model. We therefore advocate for a more informative evaluation that incorporates DEA in addition to the standard metrics. We believe this will lead to more easily interpretable results and possibly the development of better, linguistically informed SR models.

Correlation between Syntactic Complexity Metrics. Unsurprisingly, tree-based metrics have positive correlations between each other (the redish area on the right) ranging from weak to strong. Due to calculation technique overlap, some of them can show strong correlation (e.g., mean dependency distance and mean flux size).

[7]Bear in mind that using human assessments for word ordering evaluation has one downside because the assessments were collected for final sentences, and were not specifically created for word ordering evaluation. A more detailed human evaluation focused on word ordering might be needed to confirm the findings including human judgments.

team	corpus	BLEU Proj/Non-Proj	Fl_z	Ad_z	Sample sizes
AX	cs	0.25/0.19	−/−	−/−	8897/979
BinLin	cs	0.49/0.38	−/−	−/−	8897/979
AX	fi	0.25/0.2	−/−	−/−	1440/85
BinLin	fi	0.44/0.33	−/−	−/−	1440/85
OSU	fi	0.47/0.38	−/−	−/−	1440/85
AX	nl	0.28/0.2	−/−	−/−	547/138
BinLin	nl	0.39/0.3	−/−	−/−	547/138
OSU	nl	0.38/0.28	−/−	−/−	547/138
Tilburg	nl	0.43/0.36	−/−	−/−	547/138
AX	ru	0.27/0.22	−/−	−/−	5833/533
BinLin	ru	0.44/0.36	−/−	−/−	5833/533
BME-UW	hi_hdtb	0.66/0.6	−/−	−/−	1534/150
DepDist	hi_hdtb	0.66/0.62	−/−	−/−	1534/150
IMS	hi_hdtb	0.82/0.73	−/−	−/−	1534/150
LORIA	hi_hdtb	0.29/0.22	−/−	−/−	1534/150
Tilburg	hi_hdtb	0.68/0.64	−/−	−/−	1534/150
BME-UW	ko_gsd	0.54/0.38	−/−	−/−	898/91
DepDist	ko_gsd	0.51/0.37	−/−	−/−	898/91
IMS	ko_gsd	0.84/0.56	−/−	−/−	898/91
LORIA	ko_gsd	0.43/0.4	−/−	−/−	898/91
Tilburg	ko_gsd	0.08/0.06	−/−	−/−	898/91
BME-UW	ko_kaist	0.51/0.39	−/−	−/−	1849/438
IMS	ko_kaist	0.82/0.6	−/−	−/−	1849/438
LORIA	ko_kaist	0.43/0.37	−/−	−/−	1849/438
Tilburg	ko_kaist	0.14/0.11	−/−	−/−	1849/438
BME-UW	ru_syntagrus	**0.58/0.59**	**0.15/0.19**	**0.31/0.48**	6070/421
IMS	ru_syntagrus	**0.76/0.77**	0.42/0.18	0.58/0.37	6070/421
LORIA	ru_syntagrus	**0.61/0.62**	0.33/0.3	**0.39/0.55**	6070/421
Tilburg	ru_syntagrus	**0.46/0.47**	−0.2/−0.37	−0.01/−0.2	6070/421

Table 3: Median values for BLEU, Fluency, and Adequacy for projective/non-projective sentences for each submission. Medians for non-projective sentences which are higher than for the projective sentences are in bold. All comparisons were significant with $p < 0.001$. Human judgments were available for *ru_syntagrus* only.

Correlation between Syntactic Complexity and Performance Metrics. Tree-based metrics do not correlate with human assessments (ρ fluctuates around zero for median and from -0.06 to -0.29 for BME-UW).

In general no correlation between tree-based metrics and system performance was found globally (i.e., for all models and all testsets). We can use the framework to analyze results on specific corpora or languages, however. For instance, zooming in on the *fr* corpus, we can observe a weak negative correlation at the system level (correlation with the median) between tree-based metrics (e.g., $\rho = -0.38$ for mean arity and tree length) and DEA. Thus, on this corpus, performance decreases as syntactic complexity (as measured by DEA) increases. Similarly, for *ar, cs, fi, it, nl*, tree-based metrics show some negative correlation with BLEU[8] whereby ρ median values between dependency metrics and BLEU for those corpora vary from -0.21 to -0.38 for *ar*, from -0.43 to -0.57 for *cs*, from -0.2 to -0.46 for *fi*, from -0.17 to -0.34 for *it*, and from -0.29 to -0.42 for *nl*.

Such increase in correlations were observed mainly for corpora, for which performance was not high across submissions (see Mille et al. (2018)). We hypothesize that BLEU correlates more with the tree-based metrics if system performance is bad.

Significance Testing. Overall, across submissions, coefficients were found non-significant only when they were close to zero (see Figure 2b).

[8]Unfortunately no human evaluations were available for those corpora.

5.2 Projectivity

Table 3 shows performance results with respect to the projectivity parameter.

Zooming in on the *ru_syntagrus* corpus and two models, one that can produce non-projective trees, BME-UW (Kovács et al., 2019), and one that cannot, the IMS system (Yu et al., 2019), we observe two opposite trends.

For the BME-UW model, the median values for fluency and adequacy are higher for non-projective sentences. Fluency medians (proj/non-proj) are 0.15/0.19 (Mann–Whitney $U = 4109131.0$, $n_1 = 6070$, $n_2 = 421$, $p < 0.001$ two-tailed); adequacy medians (proj/non-proj) are 0.31/0.48 ($U = 2564235.0$, $n_1 = 6070$, $n_2 = 421$, $p < 0.001$). In other words, while the model can handle non-projective structures, a key drawback revealed by our error analysis is that for sentences with projective structures (which incidentally, are much more frequent in the data), the model output is in fact judged less fluent and less adequate by human annotators than for non-projective sentences.

Conversely, for the IMS system, median values for fluency is higher for projective sentences (0.42 vs. 0.18 for non-projective sentences), and the distributions in the two groups differed significantly ($U = 4038434.0$, $p < 0.001$ two-tailed). For adequacy, the median value for projective sentences (0.58) is also significantly higher than that for non-projective sentences (0.37, $U = 2583463.0$, $p < 0.001$ two-tailed). This in turn confirms the need for models that can handle non-projective structures.

Another interesting point highlighted by the results on the *ru_syntagrus* corpus in Table 3 is that similar BLEU scores for projective and non-projective structures do not necessarily mean similar human evaluation scores.

In terms of BLEU only, that is, taking all other corpora with no human evaluations, and modulo the caveat just made about the relation between BLEU and human evaluation, we find that non-projective median values were always lower than projective ones, and distributions showed significant differences, throughout all the 25 comparisons made. This underlines the need for models that can handle both projective and non-projective structures.

5.3 Entropy

Correlation between dependency relation entropy and dependency edge accuracy permits identifying which model, language, or corpus is particularly affected by word order freedom.

For instance,[9] for the *id_gsd* corpus, three teams have a Spearman's ρ in the range from -0.62 to -0.67, indicating that their model underperforms for dependency relations with free word order. Conversely, two other teams showed weak correlation ($\rho = -0.31$ and $\rho = -0.36$) for the same *id_gsd* corpus.

The impact of entropy also varies depending on the language, the corpus, and, more generally, the entropy of the data. For instance, for Japanese (*ja_gsd* corpus), dependency relations have low entropy (the mean entropy averaged on all relations is 0.02) and so we observe no correlation between entropy and performance. Conversely, for Czech (the treebank with the highest mean entropy, $H = 0.52$), two teams show non-trivial negative correlations ($\rho = -0.54$ and $\rho = -0.6$) between entropy and DEA.

5.4 Which Syntactic Constructions Are Harder to Handle?

DEA. For a given dependency relation, DEA assesses how well a model succeeds in realizing that relation. To identify which syntactic constructs are problematic for surface realization models, we therefore compute dependency edge accuracy per relation, averaging over all submissions. Table 4 shows the results.

Unsurprisingly, relations with low counts (first five relations in the table) have low accuracy. Because they are rare (in fact they are often absent from most corpora), SR models struggle to realize these.

Other relations with low accuracy are either relations with free word order (i.e., *advcl, discourse, obl, advmod*) or whose semantics is vague (*dep*—unspecified dependency). Clearly, in case of the latter, systems cannot make a good prediction; as for the former, the low DEA score may be an artefact of the fact that it is computed with respect to a single reference. As the construct may occur in different positions in a sentence,

[9] As indicated in Section 4, we computed correlation scores between entropy for all systems, all corpora and all performance scores. These are not shown here as space is lacking.

deprel	count	Accuracy
list	4,914	17.75
vocative	974	21.91
dislocated	7,832	23.11
reparandum	33	27.27
goeswith	1,453	27.98
parataxis	27,484	28.76
dep	14,496	29.80
advcl	60,719	32.52
csubj	8,229	36.60
discourse	3,862	37.45
ccomp	33,513	41.74
obl	232,097	42.39
appos	35,781	43.59
advmod	180,678	44.84
iobj	16,240	44.96
conj	149,299	45.77
orphan	843	48.49
expl	10,137	50.90
acl	79,168	51.24
cop	45,187	51.78
nsubj	268,686	51.80
xcomp	36,633	56.12
obj	190,140	57.87
nummod	61,459	58.46
aux	95,748	58.47
mark	105,993	59.77
compound	82,314	59.99
nmod	357,367	60.94
flat	62,686	61.28
amod	246,733	61.68
cc	123,866	61.94
clf	1,668	67.47
fixed	27,978	73.08
det	280,978	73.51
case	465,583	74.15

Table 4: Macro-average dependency edge accuracy over all submissions sorted from the lowest accuracy to the highest. Count is a number of times a relation was found in all treebanks.

several equally correct sentences may match the input but only one will not be penalised by the comparison with the reference. This underlines once again the need for an evaluation setup with multiple references.

Relations with the highest accuracy are those for function words (*case*—case-marking elements, *det*—determiners, *clf*—classifiers), fixed multiword expressions (*fixed*), and nominal dependents (*amod*, *nmod*, *nummod*). Those dependencies on average have higher stability with respect to their head in terms of distance, more often demonstrate a fixed word order, and do not exhibit a certain degree of probable shifting as the relations described above. Due to those factors, their realization performance is higher.

rank	subtree	cov.	MSS
1–2	(conj (X))	70–73	1.17
3	(advcl (nsubj))	62	0.91
4	(advcl (advmod))	62	0.95
5	(advmod (advmod))	59	0.77
6	(conj (advcl))	57	0.75
7	(nsubj (conj))	56	0.68
8–11	(conj (X))	52–56	0.87
12	(nmod (advmod))	52	0.56
13	(nsubj (amod))	52	0.75
14–15	(conj (X))	49–50	0.73
16	(parataxis (nsubj))	49	0.75
17	(conj (advmod advmod))	48	0.65
18	(advcl (cop))	48	0.60
19	(advcl (aux))	47	0.59
20	(ccomp (advmod))	47	0.68

Table 5: Top-20 of the most frequent suspicious trees (dep-based) across all submissions. In case of *conj*, when tree patterns were similar, they were merged, X serving as a placeholder. Coverage: percentage of submissions where a subtree was mined as suspicious. MSS: mean suspicion score for a subtree.

Interestingly, when computing DEA per dependency relation and per corpus, we found similar DEA scores for all corpora. That is, dependency relations have consistently low/high DEA score across all corpora therefore indicating that improvement on a given relation will improve performance on all corpora/languages.

Finally, we note that, at the model level, DEA scores are useful metrics for researchers as it brings interpretability and separation into error type subcases.

Error Mining for Syntactic Trees. We can also obtain a more detailed picture of which syntactic constructs degrade performance using error mining. After running error mining on all submissions, we examine the subtrees in the input that have highest coverage, that is, for which the percentage of submissions tagging these forms as suspicious[10] is highest. Tables 5, 6, and 7 show the results when using different views of the data (i.e., focusing only on dependency information, only on POS tags, or on both).

[10] A form is suspicious if its suspicion score is not null.

tree	coverage	MSS
(ADJ (PRON))	70	0.90
(VERB (VERB))	69	1.21
(ADJ (ADJ))	68	0.89
(NOUN (ADV))	67	1.03
(ADJ (ADP))	66	0.77
(VERB (ADJ))	65	0.98
(ADV (ADV))	63	0.87
(NOUN (AUX))	62	0.90
(ADJ (VERB))	60	0.80
(VERB (CCONJ))	60	1.02
(PRON (ADP))	56	0.81
(VERB (VERB VERB))	55	0.89
(NUM (NUM))	55	0.72
(PROPN (NOUN))	53	0.79
(PRON (VERB))	53	0.63
(ADJ (CCONJ))	52	0.65
(VERB (ADV))	52	0.96
(ADJ (SCONJ))	52	0.62
(VERB (ADP))	51	0.76
(VERB (PROPN))	51	0.83

Table 6: Most frequent suspicious trees (POS-based) across all submissions.

subtree	cov.	MSS
(VERB∼ conj (ADV∼advmod))	60	0.90
(VERB∼conj (PRON∼nsubj))	60	0.78
(NOUN∼nsubj (ADJ∼amod))	55	0.77
(ADV∼advmod (ADV∼advmod))	54	0.69
(VERB∼advcl (ADV∼advmod))	53	0.76
(VERB∼advcl (NOUN∼nsubj))	53	0.70
(VERB∼conj (VERB∼advcl))	50	0.60
(VERB∼advcl (PRON∼obj))	48	0.53
(VERB∼ccomp (ADV∼advmod))	47	0.57
(NOUN∼nsubj (NOUN∼conj))	46	0.46
(VERB∼advcl (NOUN∼obl))	46	0.68
(VERB∼conj (PRON∼obj))	45	0.57
(VERB∼advcl (AUX∼aux))	44	0.56
(VERB∼conj (AUX∼aux))	41	0.59
(NOUN∼obl (ADJ∼amod))	40	0.62
(NOUN∼nsubj (VERB∼acl))	40	0.46
(VERB∼acl (ADV∼advmod))	40	0.47
(NOUN∼obl (ADV∼advmod))	38	0.43
(NOUN∼conj (VERB∼acl))	38	0.38
(VERB∼ccomp (AUX∼aux))	38	0.48

Table 7: Most frequent suspicious trees (dep-POS-based) across all submissions.

Table 5 highlights coordination (*conj*, 13 subtrees out of 20) and adverbial clause modifiers (*advcl*, 5 cases) as a main source of low BLEU scores. This mirrors the results shown for single dependency relations (cf. Section 5.4) but additionally indicates specific configurations in which these relations are most problematic such as for instance, the combination of an adverbial clause modifier with a nominal subject (*nsubj*, 62% coverage), or an adverbial modifier (*advmod*, 62% coverage), or the combination of two adverbial modifiers together (e.g., *down there, far away, very seriously*).

Table 6 shows the results for the POS setting. Differently from the dep-based view, it highlights head-dependent constructs with identical POS tags, for example, (ADV (ADV)), (ADJ (ADJ)), (NUM (NUM)), (VERB (VERB)), and (VERB (VERB VERB)), as a frequent source of errors. For instance, the relative order of two adjectives (ADJ (ADJ)) is sometimes lexically driven and therefore difficult to predict (Malouf, 2000).

Table 7 shows a hybrid POS-dep view of the most suspicious forms on a system level, detailing the POS tags most commonly associated with the dependency relations shown in Table 5 to raise problem, namely, coordination, adverbial modifiers, and adverbial clauses.

6 Using Error Analysis for Improving Models or Datasets

As shown in the preceding section, the error analysis framework introduced in Section 3 can be used by evaluation campaign organizers to provide a linguistically informed interpretation of campaign results aggregated over multiple system runs, languages or corpora.

For individual researchers and model developers, our framework also provides a means to have a fine-grained interpretation of their model results that they can then use to guide model improvement, to develop new models, or to improve training data. We illustrate this point by giving some examples of how the toolkit could be used to help improve a model or a dataset.

Data Augmentation. Augmenting the training set with silver data has repeatedly been shown to increase performance (Konstas et al., 2017; Elder and Hokamp, 2018). In those approaches, performance is improved by simply augmenting the size of the training data. In contrast, information from the error analysis toolkit could be used to support error-focused data augmentation, that is, to specifically augment the training data with instances of those cases for which the model underperforms (e.g., for

dependency relations with low dependency edge accuracy, for constructions with low suspicion score or for input trees with large depth, length or mean dependency distance). This could be done either manually (by annotating sentences containing the relevant constructions) or automatically by parsing text and then filtering for those parse trees which contain the dependency relations and subtrees for which the model underperforms. For those cases where the problematic construction is frequent, we conjecture that this might lead to a better overall score increase than ''blind'' global data augmentation.

Language Specific Adaptation. Languages exhibit different word order schemas and have different ways of constraining word order. Error analysis can help identify which language-specific constructs impact performance and how to improve a language-specific model with respect to these constructs.

For instance, a dependency relation with high entropy and low accuracy indicates that the model has difficulty learning the word order freedom of that relation. Model improvement can then target a better modelling of those factors which determine word order for that relation. In Romance languages, for example, adjectives mostly occur after the noun they modify. However, some adjectives are pre-posed. As the pre-posed adjectives rather form a finite set, a plausible way to improve the model would be to enrich the input representation by indicating for each adjective whether it belongs to the class of pre- or post-posed adjectives.

Global Model Improvement. Error analysis can suggest direction for model improvement. For instance, a high proportion of non-projective sentences in the language reference treebank together with lower performance metrics for those sentences suggests improving the ability of the model to handle non-projective structures. Indeed, Yu et al. (2020) showed that the performance of the model of Yu et al. (2019) could be greatly improved by extending it to handle non-projective structures.

Treebank Specific Improvement. Previous research has shown that treebanks contain inconsistencies thereby impacting both learning and evaluation (Zeman, 2016).

The tree-based metrics and the error mining techniques provided in our toolkit can help identify those dependency relations and constructions which have consistently low scores across different models or diverging scores across different treebanks for the same language. For instance, a case of strong inconsistencies in the annotation of multi-word expressions (MWE) may be highlighted by a low DEA for the *fixed* dependency relation (which should be used to annotate MWE). Such annotation errors could also be detected using lemma-based error mining, namely, error mining for forms decorated with lemmas. Such mining would then show that the most suspicious forms are decorated with multi-word expressions (e.g., ''in order to'').

Ensemble Model. Given a model M and a test set T, our toolkit can be used to compute, for each dependency relation d present in the test set, the average DEA of that model for that relation (DEA^d_M, the sum of the model's DEA for all d-edge in T normalized by the number of these edges). This could be used to learn an ensemble model which, for each input, outputs the sentence generated by the model whose score according to this metric is highest. Given an input tree t consisting of a set of edges D, the score of a model M could for instance be the sum of the model's average DEA for the edges contained in the input tree normalized by the number of edges in that tree, namely, $\frac{1}{|D|} \times \sum_{d \in D} DEA^d_M$.

7 Conclusion

We presented a framework for error analysis that supports a detailed assessment of which syntactic factors impact the performance of surface realisation models. We applied it to the results of two SR shared task campaigns and suggested ways in which it could be used to improve models and datasets for shallow surface realisation. More generally, we believe that scores such as BLEU and, to some extent, human ratings do not provide a clear picture of the extent to which SR models can capture the complex constraints governing word order in the world natural languages. We hope that the metrics and tools gathered in this evaluation toolkit can help address this issue.

Acknowledgments

We are grateful to Kim Gerdes for sharing his thoughts at the initial stage of this research project and giving us useful literature pointers, and we thank Shashi Narayan for making his tree error mining code available to us. This research project would not also be possible without the data provided by the Surface Realization shared task organisers, whose support and responsiveness we gratefully acknowledge. We also thank our reviewers for their constructive and valuable feedback. This project was supported by the French National Research Agency (Gardent; award ANR-20-CHIA-0003, XNLG ''Multilingual, Multi-Source Text Generation'').

References

Anja Belz, Mike White, Dominic Espinosa, Eric Kow, Deirdre Hogan, and Amanda Stent. 2011. The first surface realisation shared task: Overview and evaluation results. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 217–226. Association for Computational Linguistics.

Aoife Cahill. 2009. Correlating human and automatic evaluation of a German surface realiser. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 97–100, Suntec, Singapore. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1667583.1667615, **PMID:** 19468038

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. 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 552–562, Hong Kong, China. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D19-1052

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367. **DOI:** https://doi.org/10.3115/v1/W14-3346

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/K17-2001

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the TAC 2008 update summarization task. In *Proceedings of the First Text Analysis Conference, TAC 2008*, Gaithersburg, Maryland, USA, November 17-19, 2008. NIST.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 45–51, Berlin, Germany. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P16-2008

William Dyer. 2019. Weighted posets: Learning surface order from dependency trees. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 61–73, Paris, France. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/W19-7807

Henry Elder, Jennifer Foster, James Barry, and Alexander O'Connor. 2019. Designing a symbolic intermediate representation for neural surface realization. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 65–73, Minneapolis, Minnesota. Association for Computational Linguistics. **DOI:**

https://doi.org/10.18653/v1/W19-2308 **PMCID:** PMC6981808

Henry Elder and Chris Hokamp. 2018. Generating high-quality surface realizations using data augmentation and factored sequence models. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 49–53, Melbourne, Australia. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/W18-3606

Katja Filippova and Michael Strube. 2009. Tree linearization in English: Improving language model based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 225–228, Boulder, Colorado. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1620853.1620915

Richard Futrell, Kyle Mahowald, and Edward Gibson. 2015. Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, Uppsala, Sweden. Uppsala University, Uppsala, Sweden.

Claire Gardent and Shashi Narayan. 2012. Error mining on dependency trees. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 592–600, Jeju Island, Korea. Association for Computational Linguistics.

Jesús Giménez and Lluís Màrquez. 2009. On the robustness of syntactic and semantic features for automatic MT evaluation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 250–258, Athens, Greece. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1626431.1626479

Kyle Gorman, Arya D. McCarthy, Ryan Cotterell, Ekaterina Vylomova, Miikka Silfverberg, and Magdalena Markowska. 2019. Weird inflects but OK: Making sense of morphological generation errors. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 140–151, Hong Kong, China. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/K19-1014

Kristina Gulordava and Paola Merlo. 2016. Multi-lingual dependency parsing evaluation: A large-scale analysis of word order properties using artificial data. *Transactions of the Association for Computational Linguistics*, 4:343–356. **DOI:** https://doi.org/10.1162/tacl_a_00103

Eduard Hovy, Chin-Yew Lin, and Liang Zhou. 2005. Evaluating DUC 2005 using basic elements. In *Proceedings of the 5th Document Understanding Conference (DUC)*.

Sylvain Kahane, Chunxiao Yan, and Marie-Amélie Botalla. 2017. What are the limitations on the flux of syntactic dependencies? Evidence from UD treebanks. In *Proceedings of the Fourth International Conference on Dependency Linguistics (Depling 2017)*, pages 73–82, Pisa, Italy. Linköping University Electronic Press.

Rahul Katragadda. 2009. On alternative automated content evaluation measures. In *Proceedings of the Second Text Analysis Conference*, Gaithersburg, Maryland, USA.

David King and Michael White. 2018. The OSU realizer for SRST '18: Neural sequence-to-sequence inflection and incremental locality-based linearization. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 39–48, Melbourne, Australia. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/W18-3605

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver,

Canada. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P17-1014

Ádám Kovács, Evelin Ács, Judit Ács, Andras Kornai, and Gábor Recski. 2019. BME-UW at SRST-2019: Surface realization with interpreted regular tree grammars. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 35–40, Hong Kong, China. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D19-6304 **PMID:** 30739462 **PMCID:** PMC7044605

Wei Li. 2015. Abstractive multi-document summarization with semantic information extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1908–1913, Lisbon, Portugal. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D15-1219 **PMCID:** PMC4665338

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, Michigan. Association for Computational Linguistics.

Haitao Liu. 2008. Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191.

Haitao Liu. 2010. Dependency direction as a means of word-order typology: A method based on dependency treebanks. *Lingua*, 120(6):1567–1578. **DOI:** https://doi.org/10.17791/jcs.2008.9.2.159

Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. 2012. Fully automatic semantic MT evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 243–252, Montréal, Canada. Association for Computational Linguistics.

Robert Malouf. 2000. The order of prenominal adjectives in natural language generation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 85–92, Hong Kong. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1075218.1075230

Dennis N. Mehay and Chris Brew. 2007. Bleuâtre: Flattening syntactic dependencies for MT evaluation. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 122–131.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (SR'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/W18-3601

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The second multilingual surface realisation shared task (SR'19): Overview and evaluation results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 1–17, Hong Kong, China. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D19-6301

George A. Miller. 1956. The magical number seven plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97. **DOI:** https://doi.org/10.1037/h0043158

Amit Moryossef, Yoav Goldberg, and Ido Dagan. 2019. Step-by-step: Separating planning from realization in neural data-to-text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2267–2277, Minneapolis, Minnesota. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 2012. Error mining with suspicion trees: Seeing the forest for the trees. In *Proceedings of COLING 2012*, pages 2011–2026, Mumbai, India. The COLING 2012 Organizing Committee.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Marie Candito, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Fabricio Chalub, Jinho Choi, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşikara, Hiroshi Kanayama, Jenna Kanerva, Natalia Kotsyba, Simon Krek, Veronika Laippala, Phuong Lê Hồng, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Čatălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2017. Universal dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D17-1238

Karolina Owczarzak. 2009. DEPEVAL(summ): Dependency-based evaluation for automatic summaries. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 190–198, Suntec, Singapore. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1687878.1687907

Karolina Owczarzak, Josef van Genabith, and Andy Way. 2007. Dependency-based automatic evaluation for machine translation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 80–87, Rochester, New York. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/1626281.1626292

Ratish Puduppully, Yue Zhang, and Manish Shrivastava. 2016. Transition-based syntactic linearization with lookahead features. In *Proceedings of the 2016 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 488–493, San Diego, California. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/N16-1058

Yevgeniy Puzikov, Claire Gardent, Ido Dagan, and Iryna Gurevych. 2019. Revisiting the binary linearization technique for surface realization. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 268–278, Tokyo, Japan. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/W19-8635

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1(1):81–106.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401. **DOI:** https://doi.org/10.1162/coli_a_00322

Allen Schmaltz, Alexander M. Rush, and Stuart Shieber. 2016. Word ordering without syntax. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2319–2324, Austin, Texas. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D16-1255

Anastasia Shimorina and Claire Gardent. 2019. Surface realisation using full delexicalisation. 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 3086–3096, Hong Kong, China. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D19-1305

Linfeng Song, Yue Zhang, and Daniel Gildea. 2018. Neural transition-based syntactic linearization. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 431–440, Tilburg University, The Netherlands. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/W18-6553, **PMCID:** PMC6219880

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19, 2005, Proceedings*, pages 341–351. **DOI:** https://doi.org/10.1007/978-3-540-30586-6_38

Stephen Tratz and Eduard H. Hovy. 2009. BEwT-E for TAC 2009's AESOP task. In *Proceedings of the Second Text Analysis Conference*. Gaithersburg, Maryland, USA.

Michael White and Rajakrishnan Rajkumar. 2012. Minimal dependency length in realization ranking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 244–255, Jeju Island, Korea. Association for Computational Linguistics.

Hui Yu, Xiaofeng Wu, Jun Xie, Wenbin Jiang, Qun Liu, and Shouxun Lin. 2014. RED: A reference dependency based MT evaluation metric. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2042–2051, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Xiang Yu, Agnieszka Falenska, Marina Haid, Ngoc Thang Vu, and Jonas Kuhn. 2019. IMSurReal: IMS at the surface realization shared task 2019. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 50–58, Hong Kong, China. Association for Computational Linguistics.

Xiang Yu, Simon Tannert, Ngoc Thang Vu, and Jonas Kuhn. 2020. Fast and accurate non-projective dependency tree linearization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1451–1462, Online. Association for Computational Linguistics.

Daniel Zeman. 2016. Universal annotation of Slavic verb forms. *The Prague Bulletin of Mathematical Linguistics*, 105:143–193.

Yue Zhang. 2013. Partial-tree linearization: Generalized word ordering for text synthesis. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2232–2238. AAAI Press. **DOI:** https://doi.org/10.1515/pralin-2016-0007

Yue Zhang, Graeme Blackwood, and Stephen Clark. 2012. Syntax-based word ordering incorporating a large-scale language model. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 736–746, Avignon, France. Association for Computational Linguistics.

Yue Zhang and Stephen Clark. 2015. Discriminative syntax-based word ordering for text generation. *Computational Linguistics*, 41(3):503–538. **DOI:** https://doi.org/10.1162/COLI_a_00229