Mod-D2T: A Multi-layer Dataset for Modular Data-to-Text Generation

Simon Mille
ADAPT, Dublin City University
simon.mille@adaptcentre.ie

François Lareau
OLST, Université de Montréal
francois.lareau@umontreal.ca

Stamatia Dasiopoulou
Independent Researcher
stamatia.dasiopoulou@gmail.com

Anya Belz
ADAPT, Dublin City University
anya.belz@adaptcentre.ie

Abstract

Rule-based text generators lack the coverage and fluency of their neural counterparts, but have two big advantages over them: (i) they are entirely controllable and do not hallucinate; and (ii) they can fully explain how an output was generated from an input. In this paper we leverage these two advantages to create large and reliable synthetic datasets with multiple human-intelligible intermediate representations. We present the Modular Data-to-Text (Mod-D2T) Dataset which incorporates ten intermediate-level representations between input triple sets and output text; the mappings from one level to the next can broadly be interpreted as the traditional modular tasks of an NLG pipeline. We describe the Mod-D2T dataset, evaluate its quality via manual validation and discuss its applications and limitations. Data, code and documentation are available at https://github.com/mille-s/Mod-D2T.

1 Introduction

Multi-level linguistic representations are inherent to several linguistic formalisms, including Lexical Functional Grammar (Dalrymple, 2001) and Meaning-Text Theory (Mel'čuk, 1973). They have been widely used in Natural Language Understanding (NLU), e.g. in Enhanced Universal Dependencies (Schuster and Manning, 2016) and Prague DT (Bejček et al., 2013), as well as in Natural Language Generation (NLG), e.g. in Enhanced WebNLG (Castro Ferreira et al., 2018), SRST (Mille et al., 2018), and Wikifluent (Kasner and Dusek, 2022). Reference architectures have been proposed that define modules and/or levels of representation, with the first theoretical architecture probably dating back to the work of Žolkovskij and Mel'čuk (1965), while a widely accepted applied NLG architecture was described by Reiter and Dale (1997). While end-to-end generators are very efficient (Dušek et al., 2018; Castro Ferreira et al., 2020), there is evidence that splitting the generation process into sub-steps can lead to improvements (Castro Ferreira et al., 2019; Moryossef et al., 2019; Puduppully and Lapata, 2021; Kasner and Dusek, 2022). However, corresponding datasets with multiple intermediate representational levels are scarce.

In this paper, we present the Modular Data-to-Text (Mod-D2T) dataset, which comprises the inputs from the WebNLG 2020 shared task data (Castro Ferreira et al., 2020), paired with new output texts and ten intermediate-level representations that incrementally specify the output. The mappings from one level to the next can broadly be interpreted as the modular tasks of linguistic structuring, text planning/sentence aggregation, lexicalisation, communicative structure determination, deep sentence structuring, surface sentence structuring, surface aggregation, referring expression generation (REG), linearisation/morphology resolution and surface form generation. Unlike existing multilevel datasets, which were created by adding annotated layers on top of existing text, we leverage the FORGe rule-based pipeline generator (Mille et al., 2019b) to produce multiple human-intelligible intermediate (semantic, syntactic, morphological) and final (text) representations starting from abstract structures.

2 The Mod-D2T Dataset

In this section, we describe the dataset and how it was built. Table 1 lists the 10 intermediate levels of representation with associated tasks and approximate[1] correspondence to Reiter and Dale (1997). All examples in this section are for the same output text: *103 Colmore Row, designed by John Madin, is in Birmingham. It has 23 floors and was completed in 1976.*

[1] Our Surface sentence structuring spans Reiter and Dale (1997)'s Lexicalisation, REG and Linguistic realisation.

Reiter&Dale Tasks	Mod-D2T Tasks	Mod-D2T Input	Mod-D2T Output
Content determination	—	—	—
Discourse planning	Linguistic structuring	WebNLG	PredArg
Sentence aggregation	Text planning*	PredArg	PredArg-Agg
Lexicalisation	Lexicalisation Comm. structuring Deep sent. structuring Surf. sent. structuring Synt. aggregation*	PredArg(-Agg) PredArg-Lex PredArg-Th DSynt SSynt	PredArg-Lex PredArg-Th DSynt SSynt SSynt-Agg
REG	REG*	SSynt(-Agg)	SSynt-Pro
Linguistic realisation	Word ord. and agree. resolution Surface form retrieval	SSynt(-Agg/-Pro) DMorph	DMorph SMorph

Table 1: The Mod-D2T layers (Mod-D2T Output) and tasks, and their correspondence with Reiter and Dale (1997)'s tasks; * Denotes optional modules, i.e., it is possible to generate grammatical texts without activating them.

```
<entry category="Building" eid="Id10" shape="(x (x) (x) (x) (x))"
    shape_type="sibling" size="4">
  <modifiedtripleset>
    <mtriple> 103_Colmore_Row | location | Birmingham </mtriple>
    <mtriple> 103_Colmore_Row | architect | Jonh_Madin </mtriple>
    <mtriple> 103_Colmore_Row | floorCount | 23 </mtriple>
    <mtriple> 103_Colmore_Row | completionDate | 1976 </mtriple>
  </modifiedtripleset>
</entry>
```

Figure 1: WebNLG'20 input triples.

2.1 Format

Intermediate representations in Mod-D2T are represented as CoNLL-U tables,[2] but not all CoNLL-U columns are used exactly as intended or at all. Because CoNLL-U is a linear format that we use to represent unordered graphs and trees, we delimit sentences by a <SENT> tag at the end of a group of nodes. All lines before <SENT> tag belong to the same sentence, but their relative order in the ConNLL-U file is not relevant. However, the order in which the sentences appear does correspond to their order in the text (see Table 2 for an example). For levels that are chains (in the sense explained below), the order of the lines is the order of the elements in the sentence.

2.2 Levels of representation

All ten intermediate representations in Mod-D2T are multi-sentence graphs that can be grouped into three main types: (i) **directed acyclic graphs (DAGs)** for semantic information; (ii) **unordered dependency trees** for syntactic information; and (iii) **chains** for morphological information. Nodes are connected across layers through individual IDs, and coreference is explicitly marked (see the Misc column of, e.g., Table 2). Below, we describe each level of representation in turn, showing the last DAG and the last dependency tree for our running example in full here (Tables 2–3), while the other levels are shown in Appendix C. Appendix A presents some dataset statistics, and Appendix B the tag sets used.

2.2.1 WebNLG'20 inputs

The dataset is fully aligned with the WebNLG 3.0 release,[3] in which the inputs are sets of DBpedia triples (Subject|Property|Object), as described by Gardent et al. (2017); an example is shown in Figure 1. The labels from the WebNLG properties are stored in our annotations for a one-to-one mapping between properties and linguistic sub-structures.

2.2.2 Semantic levels: DAGs

Predicate-argument graphs (PredArg) are basic predicates linked to their arguments, mainly via numbered relations in the style of PropBank (Kingsbury and Palmer, 2002), with a few exceptions for common modifiers such as time and loca-

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

[3] https://gitlab.com/shimorina/webnlg-dataset/-/tree/master/release_v3.0

ID	Lexeme	POS	Features	Head	Rel	Misc
1	design	VB	past	0	root	src=4
2	John_Madin	NP	person|ne	1	A0	src=6
3	be	VB	**rheme**	0	root	src=1
4	103_Colmore_Row	NP	ne	3,1	A1,A1	src=2|coref=0
5	Birmingham	NP	location|ne	3	A2	src=3
6	<SENT>	_	_	_	_	_
7	23	CD	_	0	root	src=9
8	have	VB	**rheme**	0	root	src=7
9	floor	NN	_	7,8	A1,A2	src=10
10	103_Colmore_Row	NP	ne	8	A1	src=8|coref=0
11	<SENT>	_	_	_	_	_
12	point_time_year	_	_	0	root	src=14
13	1976	NP	year|ne	12	A2	src=14
14	complete	VB	past|**rheme**	12	A1	src=12
15	103_Colmore_Row	NP	ne	14	A1	src=13|coref=0
16	<SENT>	_	_	_	_	_

Table 2: Predicate-argument structure with thematicity (PredArg-Th).

tion. Due to the nature of WebNLG triples, nearly all predicates at this level of representation are binary. The main difference between PredArg graphs and the RDF input is that in the former, the content is structured linguistically, in terms of language-oriented representations based on meanings and predicate/argument relations between them. Table 9 gives an example of a text with four "sentences", i.e., elementary blocks of information, corresponding to the four input triples (Figure 1).

Aggregated PredArg graphs (PredArg-Agg) represent content packaging, where predicates that have common arguments can be merged into a sentence. Table 10 shows an example: the first two sentences from the previous level have been merged into one that will express both who designed the building and where it is located.

Lexicalised PredArg graphs (PredArg-Lex) replace meanings from the previous level with specific lexical units with an associated part of speech. In multilingual generation, this is where we pivot to the target language. Table 11 shows an example.

PredArg graphs with thematicity (PredArg-Th) give communicative structure to the text, i.e., establishes what each sentence asserts (the rheme) and what it asserts it about (the theme), as proposed by Mel'čuk (2001). In practice, we usually only add a `rheme` feature to the main node of the rheme, which is essential because it identifies the syntactic root. Table 2 shows an example, where each sentence has its main node identified. Multiple predicates that share an argument (e.g. $have(103\_Colmore\_Row, floor) \wedge 23(floor)$), are represented with heads and relations separated by a comma, as on line 9.

2.2.3 Syntactic levels: Dependency trees

Deep-syntactic trees (DSynt). Based on thematicity, we establish hierarchy and introduce deep syntactic relations between the meaningful lexical units of the sentence only, mostly distinguishing between complements (numbered) and modifiers (ATTR) (Mel'čuk, 1988; Kahane, 2009). This type of tree is roughly equivalent to a non-ordered UD representation (de Marneffe et al., 2021) stripped of cases, determiners and auxiliaries. This is also where we introduce semantically motivated features such as tense and number (see Table 13).

Surface-syntactic trees (SSynt) introduce function words and surface (usually language-specific) syntactic functions *à la* Mel'čuk (1988). This type of tree is similar to Surface-Syntactic UD (Gerdes et al., 2018), minus the linearity, or to Prague dependencies (Bejček et al., 2013), minus the morphematic nodes; Table 14 shows an example.

Aggregated SSynt trees (SSynt-Agg) introduce a more surface-oriented kind of aggregation that can only be performed once the syntactic structure has been computed. For example, in Table 15, the last two sentences of the previous structure share the same Subject, so they are merged into one sentence with coordinated main clauses. This operation is intended to increase fluency.

Pronominalised SSynt trees (SSynt-Pro) introduce pronouns where linguistically needed; see Table 3. This operation is intended to increase fluency and is sometimes needed for grammaticality.

2.2.4 Word-based levels: Chains

Deep morphological chains (DMorph) form the first linear layer. It introduces agreement resolution,

ID	Lexeme	POS	Features	Head	Rel	Misc
1	be	VB	decl|fin|ind|pres|rheme	0	root	src=1
2	103_Colmore_Row	NP	sg|ne	1	SBJ	src=2|coref=0
3	design	JJ	part	2	NMOD	src=4
4	by	IN	_	3	LGS	src=6
5	John_Madin	NP	masc|sg|person|ne	4	PMOD	src=6
6	in	IN	_	1	PRD	src=3
7	Birmingham	NP	sg|location|ne	6	PMOD	src=3
8	<SENT>	_	_	_	_	_
9	have	VB	decl|fin|ind|pres|rheme	0	root	src=7
10	and	CC	_	9	COORD	src=-
11	be	VB	decl|fin|ind|past|rheme	10	CONJ	src=12
12	in	IN	_	11	ADV	src=14
13	1976	NP	year|ne	12	PMOD	src=14
14	floor	NN	pl	9	OBJ	src=10
15	23	CD	_	14	NMOD	src=9
16	**_PRO_**	**PP**	sg|ne	9	SBJ	src=8|coref=0
17	complete	VB	decl|part|rheme	11	VC	src=12
18	**_PRO_**	**PP**	sg|ne	11	SBJ	src=13|coref=0
19	<SENT>	_	_	_	_	_

Table 3: Pronominalised surface syntactic representation (SSynt-Pro).

sentence-final punctuation (typically, a period), and ellipsis earmarking, as shown in Table 17.

Surface morphological chains (SMorph) list all tokens, including non-final punctuation (parentheses, commas, etc.), together with the POS tag, source ID and coreference ID, as in Table 18.

2.3 Dataset construction

We process automatically each WebNLG'20 input triple set by running 5 modules that consecutively (1) enrich the triple set, (2) populate PredArg templates and pre-order them based on the elements they have in common, (3) generate the text while saving intermediate layers in the process, (4) clean the generated text, and (5) clean and map the intermediate representations to CoNLL-U format.

Code for steps (4-5) was created for this paper, and for steps (1-3) we use as a starting point the FORGe pipeline (Mille et al., 2019b), which we tailored to our needs as follows. For (2), to maintain alignment, we copy from the WebNLG'20 inputs the information relative to the `category` and the `eid` to each input structure, and the respective property names to each `<SENT>` of each input structure (not shown in Appendix C). For (3), we modified FORGe by (i) separating the REG submodule from the linearisation submodule it was part of, and (ii) adding functionalities to maintain node and coreference alignments across levels; we also implemented a component that enables us to call each FORGe (group of) submodule(s) separately and store our 10 intermediate representations. Our 5 modules and the produced data can be found in the following GitHub repository https://github.com/mille-s/Mod-D2T.

3 Qualitative Evaluation

We evaluated the quality of the intermediate representations by counting and classifying errors in PredArg-Th (Figure 2) and SSynt-Pro (Figure 3) for 30 randomly selected inputs of the WebNLG'20 dev split. These inputs contain 1 to 7 triples each, and are rendered as texts of up to 5 sentences.[4]

PredArg-Th: Out of the 30 semantic graphs, 20 (66.7%) were considered correct by a semantics expert (an author). The most common error was an overactive aggregation pattern that coordinated incompatible elements, typically a territory and one of its constituents, e.g., *Aarhus University is in Aarhus and Denmark*. This problem was found in 8 structures (26.7%) and it resulted in sentences that were still grammatical and faithful, but lacked fluency. We also identified one case of unnecessary predicate 'be' ('be' was used for *Abraham A. Ribicoff is American*; 'American' itself being predicative, we could do without the copula in the PredArg structure). The rest of the structures did not exhibit problems, but some representations could raise debate, such as (i) the use of "phantom

[4] For the random sample, we gave an equal probability to inputs of every size, although inputs of different sizes are not equally represented in the dataset: there are 10 to 20 times less 6- and 7-triple inputs than smaller sized ones. Large inputs are more challenging, and over-representing them as we did allowed us to detect more potential issues, but it should be noted that the numbers reported here are probably worse than they would be with a representative sample.

agents", as in *AIT is affiliated with VTU*, where the arguments are numbered A2 and A3, supposing an A1 that would affiliate one with the other (6 cases in 4 structures), and (ii) the choice of some edge labelling, e.g. for *Hypermarcas is in the pharmaceuticals sector*, which is represented with a Location relation between 'Hypermarcas' and 'sector'; a more solid representation would treat 'sector' as a predicate with 'Hypermarcas' as its argument.

SSynt-Pro: Out of the 30 syntactic structures, 28 (93%) were considered correct given the PredArg-Th representation by a syntax expert (another author). The two problems found were one case of superfluous determiner *the* next to a genitive complement (which would produce *the Baku's memorial* in the final text),[5] and one case of an underspecified dependency DEP where ADV would have been more appropriate.

Text: A previous human evaluation of the text quality of the FORGe generator that we use was provided by Castro Ferreira et al. (2020). In the "seen" scenario, they reported the following raw scores (out of 100): Data coverage: 95.3 (human-written text: 95.5), Relevance = 94.6 (94.1) , Correctness: 93.6 (93.4), Text structure: 87.0 (91.2), Fluency: 82.7 (88.1).

4 Related Work and Limitations

Moryossef et al. (2019) and Castro Ferreira et al. (2019) aligned WebNLG triples with the corresponding reference texts, making their datasets particularly appropriate for learning Reiter and Dale's Sentence aggregation. Castro Ferreira et al. (2019) also replaced the mentions of Subject and Object values with placeholders and lemmatised verbs in texts, allowing for learning both Lexicalisation as a whole and REG. Kasner and Dusek (2022) split Wikipedia paragraphs and rephrase splits into autonomous minimal sentences, replacing pronouns by their referent. They thus end up with two layers used to train a Sentence aggregation module and a paragraph compression module, which includes REG. They then apply their approach to the WebNLG dataset by crafting minimal sentence (as opposed to PredArg in our case) templates that they instantiate with the input triples. Mille et al. (2018) propose one syntactic and one predicate-argument levels using Universal Dependency annotations (de Marneffe et al., 2021) as a source.

Our dataset differs from the previous work in that we do not use human-written texts, and that we provide richer linguistic structures, with multiple semantic, syntactic and morphological levels that are (to the extent of our knowledge) not currently available for triple-to-text generation. The main limitation of our approach is that since the texts are synthetic and produced by a deterministic generator, their variety and quality is limited by the knowledge encoded in the generator (in particular, they generally lack the naturalness of human-written texts), and they represent only a fraction of what is possible for a language to express. Another (current) limitation is that Mod-D2T only contains English; generating texts in other languages requires crafting lexical resources and retrieving the Subject and Object values in the target language, adding rules to cover language-specific phenomena, and adapting PredArg templates; see (Mille et al., 2019a). We are currently completing Irish and French versions of Mod-D2T and will report on the multilingual aspects in future publications.

5 Conclusions and Future work

With the Mod-D2T dataset, we are making available a large amount of rich and reliable linguistic structures at several levels of representation for a sizeable set of D2T input/output pairs. These can be used, e.g., for experimenting with plug-and-play NLU and NLG, facilitating (human) language learning, or teaching linguistics.[6] The dataset construction process is flexible enough to allow the controlled production of a myriad of variants of the dataset in terms of verbalisation, sentence grouping/structuring, output simplicity/complexity, etc., simply by (de)activating optional modules (see Table 1) or introducing variation during the linguistic structuring task –thus providing multiple ways of verbalising each input triple. In contrast to neural generation, our approach ensures that the final text is faithful to the input, and will not contain inaccuracies, biases or offensive language.

In future work, in addition to the multilingual extension mentioned above, we will make the generation pipeline available for researchers to generate their own datasets, and provide mappings from our representations to standard representation schemes (e.g., Surface-syntactic UD).

[5] It is not the case since some rules filter out the superfluous *the* at a later stage in the pipeline.

[6] Thanks to their accuracy, the syntactic representations in particular can be used as teaching material, since unlike automatically parsed sentences, they will contain a negligible amount of errors.

Acknowledgements

Mille's contribution was funded by the European Union under the Marie Skłodowska-Curie grant agreement No 101062572 (M-FleNS).

Ethics statement

Given that we do not resort to using language models nor to human evaluation with people who are not authors of this paper, our work has no ethics implication that we are aware of.

References

Eduard Bejček, Eva Hajičová, Jan Hajič, Pavlína Jínová, Václava Kettnerová, Veronika Kolářová, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Jarmila Panevová, Lucie Poláková, Magda Ševčíková, Jan Štěpánek, and Šárka Zikánová. 2013. Prague dependency treebank 3.0.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina. 2020. The 2020 bilingual, bi-directional WebNLG+ shared task: Overview and evaluation results (WebNLG+ 2020). In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 55–76, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben. 2018. Enriching the WebNLG corpus. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 171–176, Tilburg University, The Netherlands. Association for Computational Linguistics.

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.

Mary Dalrymple. 2001. *Lexical functional grammar*, volume 34. Brill.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Computational Linguistics*, 47(2):255–308.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328, Tilburg University, The Netherlands. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Kim Gerdes, Bruno Guillaume, Sylvain Kahane, and Guy Perrier. 2018. SUD or surface-syntactic Universal Dependencies: An annotation scheme near-isomorphic to UD. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 66–74, Brussels, Belgium. Association for Computational Linguistics.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 105–112.

Sylvain Kahane. 2009. Defining the deep syntactic structure: How the signifying units combine. In *Proceedings of the Meaning-Text Conference (MTT)*, Montréal, Canada.

Zdeněk Kasner and Ondrej Dusek. 2022. Neural pipeline for zero-shot data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3914–3932, Dublin, Ireland. Association for Computational Linguistics.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Igor A. Mel'čuk. 1973. Towards a linguistic 'Meaning ↔ Text' model. *Trends in Soviet theoretical linguistics*, pages 33–57.

Igor A. Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany, NY.

Igor A. Mel'čuk. 2001. *Communicative organization in natural language: the semantic-communicative structure of sentences*. John Benjamins, Amsterdam/ Philadelphia.

Simon Mille, Anja Belz, Bernd Bohnet, and Leo Wanner. 2018. Underspecified Universal Dependency structures as inputs for multilingual surface realisation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 199–209, Tilburg University, The Netherlands. Association for Computational Linguistics.

Simon Mille, Stamatia Dasiopoulou, Beatriz Fisas, and Leo Wanner. 2019a. Teaching FORGe to verbalize DBpedia properties in Spanish. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 473–483, Tokyo, Japan. Association for Computational Linguistics.

Simon Mille, Stamatia Dasiopoulou, and Leo Wanner. 2019b. A portable grammar-based nlg system for verbalization of structured data. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1054–1056.

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.

Ratish Puduppully and Mirella Lapata. 2021. Data-to-text generation with macro planning. *Transactions of the Association for Computational Linguistics*, 9:510–527.

Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2371–2378, Portorož, Slovenia. European Language Resources Association (ELRA).

Aleksandr K. Žolkovskij and Igor A. Mel'čuk. 1965. O vozmožnom metode i instrumentax semantičeskogo sinteza [Method and instruments of semantic synthesis]. *Naučnotexničeskaja informacija*, 5:23–28.

A Statistics

There are 13,211, 1,667 and 1,779 texts in the training, development and test splits respectively. Tables 4-5 provide an overview of the number of nodes and sentences per text for all splits. Our 10 intermediate layers contain over 1.9 million nodes.

B Tagsets used

The edge labels for semantic graphs come mainly from PropBank (Kingsbury and Palmer, 2002), plus some generic labels such as `Location` and `Time`; see Table 6. The ones for deep syntactic trees come from Meaning-Text Theory (Mel'čuk, 1988); see Table 7. As for surface syntactic edge labels, they are a subset of the dependency Penn Treebank labels (Johansson and Nugues, 2007); see Table 8.

Layer	N	S
PredArg	152,664	48,776
PredArg-Agg	134,188	31,204
PredArg-Lex	134,188	31,204
PredArg-Comm	143,448	31,204
DSynt	169,325	31,204
SSynt	219,962	31,204
SSynt-Agg	222,970	27,557
REG	220,218	27,557
DMorph	247,795	27,557
Text	268,267	27,557

Table 4: Total number of nodes (N) and sentences (S) per layer.

Layer	N	S	N/S
PredArg	9.2	2.9	3.1
PredArg-Agg	8.1	1.9	4.4
PredArg-Lex	8.1	1.9	4.4
PredArg-Th	8.6	1.9	4.7
DSynt	10.2	1.9	5.5
SSynt	13.2	1.9	7.1
SSynt-Agg	13.4	1.7	8.2
SSynt-Pro	13.2	1.7	8.1
DMorph	14.9	1.7	9.1
SMorph	16.1	1.7	9.9

Table 5: Average number of nodes (N), sentences (S) and nodes per sentence (N/S) for each text, per layer.

C Sample structures

The annotations are released in CoNLL-U format, but because of space constraints, we have truncated the data in Tables 9–18 below:

- we dropped unused columns and renamed the remaining ones for readability;
- we removed feature names to retain only their values;
- we omit the metadata, which specifies the text ID (irrelevant here), the level of representation (see the captions) and the corresponding text string (see below).

The elements in bold below highlight the main changes between levels. The showcased structures all correspond to the following text:

> *103 Colmore Row, designed by John Madin, is in Birmingham. It has 23 floors and was completed in 1976.*

Label	Description	Example
A0—A6	n-th argument of a predicate or quasi-predicate	speak→ English
Location	location	born→ Paris
Time	time	build→ 1932
NonCore	inverted first argument of a predicate	runway→ second
Set	list of elements	and→ speak
Elaboration	(i) none of governor or dependent are argument of the other (ii) unknown argument slot	above me→ 610m

Table 6: Edge labels: semantic graphs

Label	Description	Example
I—VI	n-th complement of a syntactic predicate	speak→ English
ATTR	modifier	runway→ second
COORD	coordination	staff members→ and
APPEND	parenthetical modifier	Hypermarcas Brazil→ (s.a.)

Table 7: Edge labels: deep syntactic trees

Label	Description	Example
ADV	adverbial (broadly)	built→ in 1932
AMOD	argument or modifier of an adjective	similar→ to
AMOD_COMP	argument of a comparative adjective	higher→ than
COORD	between conjunct and conjunction	and→ speak
DEP	underspecified	—
EXT	prepositional object (not *to*)	ask→for
IM	infinitive marker	to→ ask
IOBJ	dative object (after OBJ)	give→ her
LGS	logical subject	owned→ by
NMOD	argument or modifier of a noun	runway→ fifth
OBJ	non-prepositional object	give→ medal
OPRD	prepositional object (*to*)	give→ to
PMOD	complement of a preposition	to→ her
PRD	predicative complement	be→ president
SBJ	syntactic subject	play→ Beatles
SUB	complement of a conjunction	while→ be

Table 8: Edge labels: surface syntactic trees

```
<entry category="Building" eid="Id10" shape="(x (x) (x) (x) (x))"
    shape_type="sibling" size="4">
  <modifiedtripleset>
    <mtriple> 103_Colmore_Row | location | Birmingham </mtriple>
    <mtriple> 103_Colmore_Row | architect | Jonh_Madin </mtriple>
    <mtriple> 103_Colmore_Row | floorCount | 23 </mtriple>
    <mtriple> 103_Colmore_Row | completionDate | 1976 </mtriple>
  </modifiedtripleset>
</entry>
```

Figure 2: WebNLG'20 input triples (same as Figure 1).

ID	Semanteme	Features	Head	Rel	Misc
1	be	_	0	root	src=1
2	Birmingham	location|ne	1	A2	src=3
3	103_Colmore_Row	ne	1	A1	src=2|coref=0
4	<SENT>	_	_	_	_
5	design	past	0	root	src=4
6	John_Madin	person|ne	5	A1	src=6
7	103_Colmore_Row	ne	5	A2	src=5|coref=0
8	<SENT>	_	_	_	_
9	have	_	0	root	src=7
10	23	_	0	root	src=9
11	floor	_	9,10	A2,A1	src=10
12	103_Colmore_Row	ne	9	A1	src=8|coref=0
13	<SENT>	_	_	_	_
14	complete	past	0	root	src=12
15	1976	year|ne	14	Time	src=14
16	103_Colmore_Row	ne	14	A2	src=13|coref=0
17	<SENT>	_	_	_	_

Table 9: Predicate-argument structure (PredArg).

ID	Semanteme	Features	Head	Rel	Misc
1	be	_	0	root	src=1
2	design	past	0	root	src=4
3	103_Colmore_Row	ne	**1,2**	**A1,A2**	src=2|coref=0
4	Birmingham	location|ne	1	A2	src=3
5	John_Madin	person|ne	2	A1	src=6
6	<SENT>	_	_	_	_
7	23	_	0	root	src=9
8	have	_	0	root	src=7
9	floor	_	7,8	A1,A2	src=10
10	103_Colmore_Row	ne	8	A1	src=8|coref=0
11	<SENT>	_	_	_	_
12	complete	past	0	root	src=12
13	1976	year|ne	12	Time	src=14
14	103_Colmore_Row	ne	12	A2	src=13|coref=0
15	<SENT>	_	_	_	_

Table 10: Aggregated predicate-argument structure (PredArg-Agg).

ID	Lexeme	**POS**	Features	Head	Rel	Misc
1	be	**VB**	_	0	root	src=1
2	Birmingham	**NP**	location|ne	1	A2	src=3
3	design	**VB**	past	0	root	src=4
4	John_Madin	**NP**	person|ne	3	A1	src=6
5	103_Colmore_Row	**NP**	ne	1,3	A1,A2	src=2|coref=0
6	<SENT>	_	_	_	_	_
7	23	**CD**	_	0	root	src=9
8	have	**VB**	_	0	root	src=7
9	floor	**NN**	_	7,8	A1,A2	src=10
10	103_Colmore_Row	**NP**	ne	8	A1	src=8|coref=0
11	<SENT>	_	_	_	_	_
12	complete	**VB**	past	0	root	src=12
13	1976	**NP**	year|ne	12	Time	src=14
14	103_Colmore_Row	**NP**	ne	12	A2	src=13|coref=0
15	<SENT>	_	_	_	_	_

Table 11: Lexicalised predicate-argument structure (PredArg-Lex).

ID	Lexeme	POS	Features	Head	Rel	Misc
1	design	VB	past	0	root	src=4
2	John_Madin	NP	person|ne	1	A0	src=6
3	be	VB	**rheme**	0	root	src=1
4	103_Colmore_Row	NP	ne	3,1	A1,A1	src=2|coref=0
5	Birmingham	NP	location|ne	3	A2	src=3
6	<SENT>	_	_	_	_	_
7	23	CD	_	0	root	src=9
8	have	VB	**rheme**	0	root	src=7
9	floor	NN	_	7,8	A1,A2	src=10
10	103_Colmore_Row	NP	ne	8	A1	src=8|coref=0
11	<SENT>	_	_	_	_	_
12	point_time_year	_	_	0	root	src=14
13	1976	NP	year|ne	12	A2	src=14
14	complete	VB	past|**rheme**	12	A1	src=12
15	103_Colmore_Row	NP	ne	14	A1	src=13|coref=0
16	<SENT>	_	_	_	_	_

Table 12: Predicate-argument structure with thematicity (PredArg-Th, same as Table 2).

ID	Lexeme	POS	Features	Head	Rel	Misc
1	be	VB	**decl|act|fin|pres**|rheme	0	root	src=1
2	Birmingham	NP	location	1	**II**	src=3
3	103_Colmore_Row	NP	_	**1**	**I**	src=2|coref=0
4	design	VB	**part**|past	**3**	**ATTR**	src=4
5	John_Madin	NP	person	4	**I**	src=6
6	<SENT>	_	_	_	_	_
7	have	VB	**decl|act|fin|pres**|rheme	0	root	src=7
8	floor	NN	**pl**	**7**	**II**	src=10
9	103_Colmore_Row	NP	_	7	**I**	src=8|coref=0
10	23	CD	_	**8**	**ATTR**	src=9
11	<SENT>	_	_	_	_	_
12	complete	VB	**decl|pass|fin**|past|rheme	0	root	src=12
13	in	IN	_	12	**ATTR**	src=14
14	1976	NP	year	13	**II**	src=14
15	103_Colmore_Row	NP	_	12	**II**	src=13|coref=0
16	<SENT>	_	_	_	_	_

Table 13: Deep syntactic representation (DSynt).

ID	Lexeme	POS	Features	Head	Rel	Misc
1	be	VB	decl|fin|**ind**|pres|rheme	0	root	src=1
2	103_Colmore_Row	NP	**sg**|ne	1	**SBJ**	src=2|coref=0
3	design	JJ	part	2	**NMOD**	src=4
4	in	IN	_	1	**PRD**	src=3
5	Birmingham	NP	**sg**|location|ne	4	**PMOD**	src=3
6	by	IN	_	3	**LGS**	src=6
7	John_Madin	NP	**masc|sg**|person|ne	6	**PMOD**	src=6
8	<SENT>	_	_	_	_	_
9	have	VB	decl|fin|**ind**|pres|rheme	0	root	src=7
10	floor	NN	**pl**	9	**OBJ**	src=10
11	23	CD		10	**NMOD**	src=9
12	103_Colmore_Row	NP	**sg**|ne	9	**SBJ**	src=8|coref=0
13	<SENT>	_	_	_	_	_
14	be	VB	decl|fin|**ind**|past|rheme	0	root	src=12
15	in	IN	_	14	**ADV**	src=14
16	1976	NP	year|ne	15	**PMOD**	src=14
17	103_Colmore_Row	NP	**sg**|ne	14	**SBJ**	src=13|coref=0
18	complete	VB	decl|**part**|rheme	14	**VC**	src=12
19	<SENT>	_	_	_	_	_

Table 14: Surface syntactic representation (SSynt).

ID	Lexeme	POS	Features	Head	Rel	Misc
1	be	VB	decl|fin|ind|pres|rheme	0	root	src=1
2	103_Colmore_Row	NP	sg|ne	1	SBJ	src=2|coref=0
3	design	JJ	part	2	NMOD	src=4
4	by	IN	_	3	LGS	src=6
5	in	IN	_	1	PRD	src=3
6	Birmingham	NP	sg|location|ne	5	PMOD	src=3
7	John_Madin	NP	masc|sg|person|ne	4	PMOD	src=6
8	<SENT>	_	_	_	_	_
9	have	VB	decl|fin|ind|pres|rheme	0	root	src=7
10	**and**	**CC**	_	**9**	**COORD**	src=-
11	be	VB	decl|fin|ind|past|rheme	10	CONJ	src=12
12	in	IN	_	11	ADV	src=14
13	1976	NP	year|ne	12	PMOD	src=14
14	103_Colmore_Row	NP	sg|ne	9	SBJ	src=8|coref=0
15	floor	NN	pl	9	OBJ	src=10
16	23	CD	_	15	NMOD	src=9
17	complete	VB	decl|part|rheme	11	VC	src=12
18	103_Colmore_Row	NP	sg|ne	11	SBJ	src=13|coref=0
19	<SENT>	_	_	_	_	_

Table 15: Aggregated surface syntactic representation (SSynt-Agg).

ID	Lexeme	POS	Features	Head	Rel	Misc
1	be	VB	decl|fin|ind|pres|rheme	0	root	src=1
2	103_Colmore_Row	NP	sg|ne	1	SBJ	src=2|coref=0
3	design	JJ	part	2	NMOD	src=4
4	by	IN	_	3	LGS	src=6
5	John_Madin	NP	masc|sg|person|ne	4	PMOD	src=6
6	in	IN	_	1	PRD	src=3
7	Birmingham	NP	sg|location|ne	6	PMOD	src=3
8	<SENT>	_	_	_	_	_
9	have	VB	decl|fin|ind|pres|rheme	0	root	src=7
10	and	CC	_	9	COORD	src=-
11	be	VB	decl|fin|ind|past|rheme	10	CONJ	src=12
12	in	IN	_	11	ADV	src=14
13	1976	NP	year|ne	12	PMOD	src=14
14	floor	NN	pl	9	OBJ	src=10
15	23	CD	_	14	NMOD	src=9
16	**_PRO_**	**PP**	sg|ne	9	SBJ	src=8|coref=0
17	complete	VB	decl|part|rheme	11	VC	src=12
18	**_PRO_**	**PP**	sg|ne	11	SBJ	src=13|coref=0
19	<SENT>	_	_	_	_	_

Table 16: Pronominalised surface syntactic representation (SSynt-Pro, same as Table 3).

ID	Word	POS	Features	Misc
1	103_Colmore_Row	NP	_	src=2|coref=0
2	design	JJ	part	src=4
3	by	IN	_	src=6
4	John_Madin	NP	_	src=6
5	be	VB	decl|fin|ind|pres|**sg**	src=1
6	in	IN	_	src=3
7	Birmingham	NP	_	src=3
8	**.**	_	_	src=-
9	_PRO_	PP	sg	src=8|coref=0
10	have	VB	decl|fin|ind|pres|**sg**	src=7
11	23	CD	_	src=9
12	floor	NN	pl	src=10
13	and	CC	_	src=-
14	_PRO_	PP	sg|**delete**	src=13|coref=0
15	be	VB	decl|fin|ind|past|**sg**	src=12
16	complete	VB	decl|part	src=12
17	in	IN	_	src=14
18	1976	NP	_	src=14
19	**.**	_	_	src=-

Table 17: Deep morphological representation (DMorph).

ID	Word	POS	Misc
1	103_Colmore_Row	NP	src=2|coref=0
2	**,**	_	src=-
3	**designed**	JJ	src=4
4	by	IN	src=6
5	John_Madin	NP	src=6
6	**,**	_	src=-
7	**is**	VB	src=1
8	in	IN	src=3
9	Birmingham	NP	src=3
10	.	_	src=-
11	**it**	PP	src=8|coref=0
12	**has**	VB	src=7
13	23	CD	src=9
14	**floors**	NN	src=10
15	and	CC	src=-
16	**was**	VB	src=12
17	**completed**	VB	src=12
18	in	IN	src=14
19	1976	NP	src=14
20	.	_	src=-

Table 18: Surface morphological representation (SMorph).