FGD at MRP 2020: Prague Tectogrammatical Graphs

Daniel Zeman and Jan Hajič
Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics (ÚFAL)
{zeman|hajic}@ufal.mff.cuni.cz

Abstract

Prague Tectogrammatical Graphs (PTG) is a meaning representation framework that originates in the tectogrammatical layer of the Prague Dependency Treebank (PDT) and is theoretically founded in Functional Generative Description of language (FGD). PTG in its present form has been prepared for the CoNLL 2020 shared task on Cross-Framework Meaning Representation Parsing (MRP). It is generated automatically from the Prague treebanks and stored in the JSON-based MRP graph interchange format. The conversion is partially lossy; in this paper we describe what part of annotation was included and how it is represented in PTG.

1 Introduction

The Functional Generative Description (FGD) (Sgall, 1967; Sgall et al., 1986), as instantiated in the Prague family of dependency treebanks, defines four layers of description: 1. the word layer; 2. the morphological layer; 3. the analytical (surface-syntactic) layer; 4. the tectogrammatical (deep-syntactic) layer. The meaning representation used in the CoNLL 2020 shared task (Oepen et al., 2020) is based mostly on the tectogrammatical layer; however, references have to be followed all the way down to the word layer in order to provide anchoring of graph nodes in the underlying text.

The shared task featured PTG data in two languages: English and Czech. The English data was taken from the same sources as in the previous shared task (CoNLL MRP 2019, Oepen et al. 2019); however, a different conversion procedure had been used in the previous task, leading to different (and simpler) target graphs, known as Prague Semantic Dependencies (PSD, Miyao et al. 2014). The source text originates in the Wall Street Journal portion of the Penn TreeBank and the source annotation stems from the Prague Czech-English Dependency Treebank 2.0 (PCEDT, Hajič et al. 2012). As there are other frameworks in which the same data is annotated in the shared task, the training-development-test split was synchronized across frameworks,[1] and a handful of sentences were omitted because they did not align with the original WSJ text. In addition, the test set includes 100 out-of-domain sentences from The Little Prince short novel by Antoine de Saint-Exupéry. The Czech data was taken from the Prague Dependency Treebank 3.5 (PDT, Hajič et al. 2017) and its standard training-development-evaluation split was used.

The meaning representation in P(CE)DT is called *tectogrammatical tree* or *t-tree*. The structure meets the tree constraints only because

- paratactic structures such as coordination are encoded using technical dependencies, special edge labels and attributes;
- coreference links are encoded as node attributes instead of being treated as edges.

As the representations in the shared task are not restricted to trees, additional edges were added to more directly encode paratactic structures and coreference. The resulting structures are called **Prague Tectogrammatical Graphs (PTG).**

2 Graph Properties and Anchoring

The typical node in a tectogrammatical graph corresponds to a content word, which is its anchoring in the surface sentence. Pronouns are treated as content words in this respect. Function words normally

[1] Sections 00–20 of WSJ served as training data; section 21 was used for development/validation and section 23 for evaluation.

DENOM	independent nominal
PAR	parenthetic clause
PARTL	independent interjection
PRED	independent verbal clause
VOCAT	independent vocative

Table 1: Five PTG functors for "independent" nodes. Except for PAR, these functors typically occur at edges going out of the artificial root node.

ACT	argument: actor
ADDR	argument: addressee
EFF	argument: effect
ORIG	argument: origo
PAT	argument: patient

Table 2: Five argument functors in PTG.

do not have nodes of their own.[2] They are treated as attributes of the content word they "belong to". This association is projected to anchoring and one node can thus be anchored to multiple surface substrings, even discontinuous. Punctuation symbols are even less prominent than function words and are not included in node anchoring.

On the other hand, there are *generated (empty) nodes* that represent reconstructed material, deleted on the surface. These nodes are usually unanchored. Unanchored is also the artificial root node. Despite not being trees, the graphs are rooted and every node is reachable from the single root node[3] via at least one directed path. Some nodes are reachable via multiple paths and the graph may also contain cycles.

In the classification of the MRP shared task, Prague Tectogrammatical Graphs represent a Flavor 1 framework.

3 Edge Types

Most edges in PTG are understood as dependencies. In each clause, the backbone of the representation is formed by edges going from a verbal predicate to its arguments and adjuncts (both of which can be clauses themselves, represented by their predicates). Copular clauses are headed by the copula (meaning that the copula is not treated as a function word) and the non-verbal component of the predicate is analyzed as an argument of the cop-

ACMP	adjunct: accompaniment
AIM	adjunct: purpose
BEN	adjunct: benefactor
CAUS	adjunct: cause
CNCS	adjunct: concession
COMPL	adjunct: predicative complement
COND	adjunct: condition
CONTRD	adjunct: confrontation
CPR	adjunct: comparison
CRIT	adjunct: criterion
DIFF	adjunct: difference
DIR1	adjunct: where from
DIR2	adjunct: which way
DIR3	adjunct: where to
EXT	adjunct: extent
HER	adjunct: inheritance
INTT	adjunct: intention
LOC	adjunct: where
MANN	adjunct: manner
MEANS	adjunct: means
REG	adjunct: with regard to
RESL	adjunct: result
RESTR	adjunct: exception, restriction
SUBS	adjunct: substitution
TFHL	adjunct: for how long
TFRWH	adjunct: from when
THL	adjunct: (after) how long
THO	adjunct: how often
TOWH	adjunct: to when
TPAR	adjunct: in parallel with what
TSIN	adjunct: since when
TTILL	adjunct: until when
TWHEN	adjunct: when

Table 3: Thirty-three adjunct functors in PTG.

APP	adnominal adjunct: appurtenance
AUTH	adnominal adjunct: author
DESCR	adnominal description (only PCEDT)
ID	adnominal specification of identity
MAT	adnominal argument: content
RSTR	adnominal adjunct: modification

Table 4: Six adnominal functors in PTG.

[2] Coordinating conjunctions (or even coordinating punctuation) are an exception. Despite being function words, they may be used as technical means to capture coordination, in which case they will have their own node.

[3] One could argue that the root node could be removed in the MRP environment and its children marked as *top nodes* instead. However, we stick to this representation because 1. the root is considered a node in the Prague tectogrammatical trees; 2. there may be multiple outgoing edges from the root, and 3. the labels and attributes of the edges are not necessarily identical. Root nodes are not labeled in the data, but in the diagrams in this paper, we use '#Root' to represent them.

ATT	speaker's attitude
CM	conjunction modifier
CPHR	nominal part of complex predicate
DPHR	dependent part of idiom
FPHR	part of foreign expression
INTF	expletive subject
MOD	some modal expressions
NE	part of named entity (only PCEDT)
PREC	preceding context
RHEM	rhematizer

Table 5: Ten PTG functors for miscellaneous dependents.

ADVS	parataxis: adversative
APPS	parataxis: apposition
CONFR	parataxis: confrontation
CONJ	parataxis: conjunction
CONTRA	parataxis: conflict
CSQ	parataxis: consequence
DISJ	parataxis: disjunction
GRAD	parataxis: gradation
OPER	parataxis: math operation
REAS	parataxis: cause

Table 6: PTG functors for 10 types of paratactic relations.

ula. There is no direct edge between the non-verbal or secondary predicate and the subject argument (Figure 1).

Unlike in some other meaning representation frameworks, attributes of nominals (and adjuncts of clauses) are not treated as predicates and the edge goes from the nominal to its attribute, not the other way around.

Overall the tectogrammatical layer defines[4] 67 relation types, called *functors*; a few extra functors are defined in PCEDT. Relations between the artificial root node and the most independent word of the sentence are listed in Table 1. Tables 2, 3 and 4 list the functors for arguments, adjuncts and adnominal modifiers, respectively. Miscellaneous other dependencies are covered by Table 5.

Optionally, some functors may be further subclassified using *subfunctors*. In the PTG data for the shared task, we merge the functor with its subfunctor into a single label, using the period

[4]See https://ufal.mff.cuni.cz/pdt2.0/doc/manuals/en/t-layer/html/ch07.html for detailed documentation.

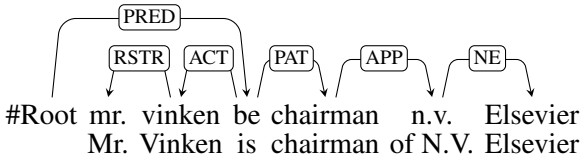

Figure 1: PTG of a simple clause with a copula. The sentence is *Mr. Vinken is chairman of Elsevier N.V.* The preposition *of* does not have a node of its own but it is considered an attribute of the head node of the named entity. The text anchoring of that node (shown in the second line) is thus *of N.V.* The linear ordering of nodes in our diagrams is not significant.

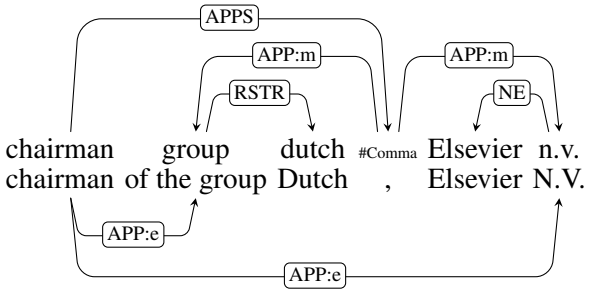

Figure 2: PTG of a paratactic structure (apposition). The phrase is *chairman of Elsevier N.V., the Dutch group*. Note that APPS means 'apposition' while APP stands for 'appurtenance'. The suffix :m in the edge labels is not a subfunctor. It is a shortcut for the *member* attribute, indicating that this node is a member of the paratactic structure, rather than its shared dependent. The edges below the nodes were added during the conversion from t-trees to PTG and connect the members of the apposition with their effective parent. The suffix :e is a shortcut for the *effective* attribute of the edge.

as a delimiter. For example, the locative adjunct LOC may be further specified as LOC.above, LOC.around etc.

Paratactic structures such as coordination and apposition call for special treatment within this prevailingly dependency-based framework. They are always headed by a technical node which is typically anchored in a coordinating conjunction or punctuation. The functor of the incoming edge only classifies the type of the paratactic relation (see Table 6 for available functors). Edges outgoing from the technical node lead to members of the paratactic relation and their functors reflect the actual relation between the parent of the structure and the member.

The technical node and the edges described so far are present also in the source t-trees in the Prague treebanks. During conversion to PTG, we

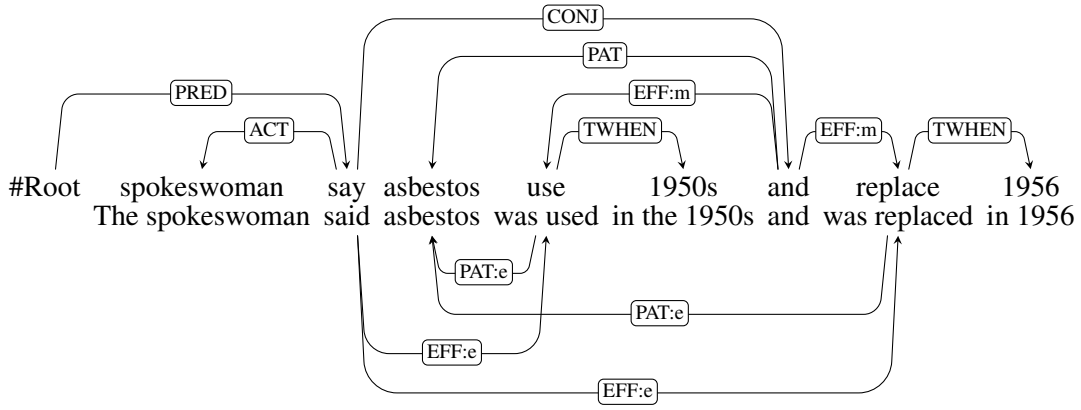

Figure 3: Propagation of effective dependencies to a shared argument of coordinate verbs. The sentence is *The spokeswoman said asbestos was used in the 1950s and replaced in 1956.* Note that the auxiliary *was* occurs only once but is included in anchoring of two nodes.

deterministically add extra edges that propagate dependencies across the paratactic structure and connect children with their *effective* parents. An example is given in Figure 2. A larger example in Figure 3 shows dependency propagation to a dependent shared by the conjuncts.

The last category of edges, also added during conversion to PTG, is related to coreference and will be described in §5.

4 Generated Nodes

Material may be missing (elided, deleted) from the surface sentence if it is unimportant or understandable from context. The tectogrammatical representation uses *generated nodes* to account for the missing material. If there is no surface word representing an obligatory valency-licensed argument of a verb, a generated node will be added and attached to the predicate with an appropriate functor. In fact, the graph in Figure 3 should include two generated nodes which we omitted for simplicity: the `ADDR` argument of *say* and the `ACT` argument of *use* and *replace*.

The labels of the generated nodes further distinguish their type and purpose. `#PersPron` is used for personal pronouns, regardless whether they are overt or generated. `#Cor` is a grammatically controlled coreferential argument (see §5), and `#Gen` is a general actor, not identifiable with a concrete entity (as in Czech *Tohle se tak prostě dělá.* "One simply does it this way."[5]

While most generated nodes are unanchored, sometimes a generated node is a copy of a regular node and inherits its anchoring (and label). Such copied nodes may be observed in coordination, as in Figure 4.

5 Coreference

Coreference is a relation between two nodes that have the same referent in the scene described by the text. While most participants in coreference are nouns or pronouns, sometimes a referent may also be described by a clause. T-trees capture coreference as a node attribute which refers to another node by its unique identifier. In PTG these links are converted to edges, as in Figure 4, where a generated node is coreferential with an overt pronoun.

While coreference is naturally a symmetric relation, only one-way direction is explicitly captured by the edge in PTG. The rules that govern the direction (inherited from the t-trees) are complex. For example, if the edge connects an overt pronoun with an overt noun, it always points from the pronoun to the noun. There may be chains of coreference edges that connect more than two coreferential nodes, and coreference edges may also cause the graph to contain cycles (Figure 5). Coreference in the Prague treebanks may even cross sentence boundaries; however, only intra-sentence relations are preserved in PTG.

There are two types of coreference edges. Grammatical coreference (`coref.gram`) follows deterministically from grammatical rules (e.g., the subject of an infinitive must be coreferential with one of the arguments of the matrix verb). The instances that do not fall under grammatical coreference are called textual coreference (`coref.text`); the

[5]The Czech sentence does not contain a word directly corresponding to the English pronoun *one*, so a `#Gen` node must be generated instead.

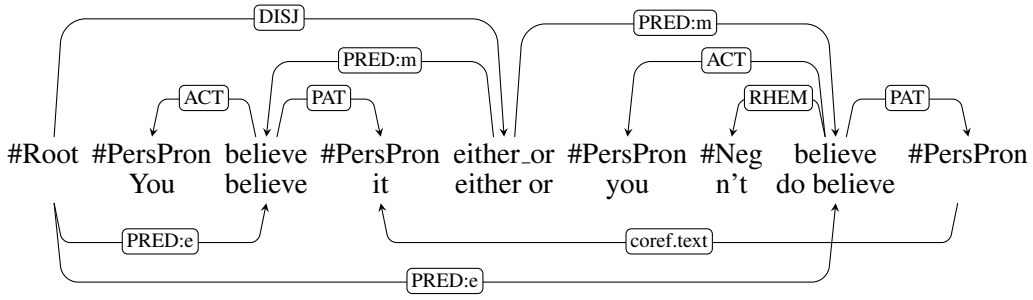

Figure 4: Ellipsis and generated nodes. The sentence is *You either believe it or you don't.* There are two nodes anchored to the surface word *believe*: the first one is a regular node, the second one is generated (in addition, the anchoring of the second node includes the auxiliary *do*). Another generated node represents the hypothetical patient of the second *believe*. The `coref.text` edge indicates that the patients of the two verbs are coreferential.

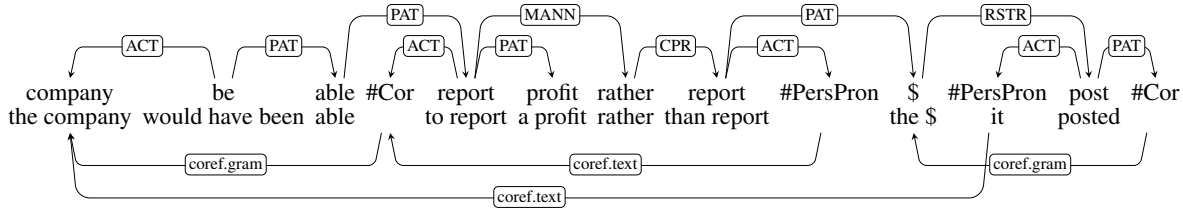

Figure 5: Coreference. The full sentence is *Without the Cray-3 research and development expenses, the company would have been able to report a profit of $19.3 million for the first half of 1989 rather than the $5.9 million it posted.* We have omitted some nodes for simplicity.

prototypical case is a pronoun linked to a noun.

Finally, the graphs may also contain edges that represent *bridging* relations. Bridging is similar to coreference but different in that the participants are not fully identical. Instead, one may be a subset of the other (then the edge label is `bridging.SUB_SET`). Bridging relations are currently available in the Czech data but not in English.

6 Node Properties

The main label that represents a node is its tectogrammatical lemma or *t_lemma*.[6] Besides it, a t-tree node has a number of attributes and 'grammatemes', both of which translate as node *properties* in the file format used in the shared task. Not all properties are available in both Czech and English, and not all properties are preserved during the conversion to PTG. The Prague treebanks, especially the Czech PDT, contain a number of grammatemes that were assigned semi-automatically without much human intervention. Such properties were omitted and only the manually assigned (or checked) ones were carried over to PTG.

The following node properties appear in the data:[7]

- **sempos** – semantic part-of-speech category. Older data, such as the English part of PCEDT, do not have sempos but they have a 'formeme', first part of which corresponds to sempos (while the second part corresponds to what newer data captures in subfunctors). The first part of a formeme is thus converted to sempos in PTG.

- **sentmod** – sentence modality, 5 values: `enunc` (declarative), `excl` (exclamative), `desid` (desiderative), `imper` (imperative), `inter` (interrogative). Occurs at the main predicate node, both in Czech and English.

- **factmod** – factual modality: is the event presented as given, or hypothetical? Four values: `asserted`, `potential`, `irreal`, `appeal`. Occurs at predicate nodes, in Czech data only.

- **diatgram** – diathesis, 7 values: `act` (active), `pas` (passive), `res1` and `res2` (resul-

[6] In the diagrams throughout this paper, the first line of each node shows its t_lemma and the second line shows the surface strings it is anchored to, if any.

[7] See `https://ufal.mff.cuni.cz/pdt3.0/documentation` for detailed documentation.

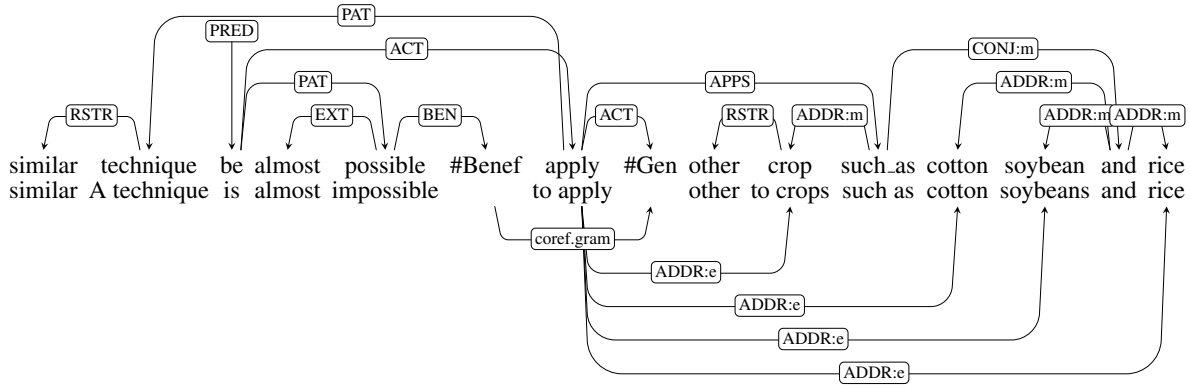

Figure 6: PTG representation of the sentence *A similar technique is almost impossible to apply to other crops, such as cotton, soybeans and rice.* (Sentence 13 of file 0209 from the Wall Street Journal / Prague Czech-English Dependency Treebank.) We have omitted the artificial #Root node in order to fit the graph on the page.

tative), `recip` (reciprocal), `disp` (dispositional), and `deagent` (deagentive). Occurs with finite verbs, in Czech data only.

- **typgroup** – does the noun in plural signify a pair/tuple? Seven possible values, e.g., `sg.group` or `pl.single`. Occurs with nouns, in Czech data only.

- **frame** – frame identifier (can be used as an index to the valency dictionary Vallex). In English, frames are available only for verbs. In Czech they are available also for some adjectives and nouns.

- **tfa** – topic-focus articulation, 3 values: `t` (topic), `f` (focus), `c` (contrast). Only in Czech data, available for most nodes (also generated ones), except the artificial root and the technical heads of paratactic structures.

Related to topic-focus articulation, the tectogrammatical layer also defines a *deep ordering* of nodes. It is reflected in the numerical node ids in Czech PTG, hence it could be considered as another node property. Note however that the diagrams in this paper do *not* reflect the deep order.

7 Other Crops

Instead of a summary, we provide in Figure 6 the PTG representation of sentence 13 of file 0209 from the Wall Street Journal, which has become a standard running example throughout many papers on semantic representations and parsing. See also Oepen et al. (2020) for an alternative but equivalent visualization of the graph.

Acknowledgments

This work was supported by the the Grant No. GX20-16819X of the Czech Science Foundation (LUSyD) and by the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101).

References

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Ondřej Bojar, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3153–3160, İstanbul, Turkey. ELRA, European Language Resources Association.

Jan Hajič, Eva Hajičová, Marie Mikulová, and Jiří Mírovský. 2017. Prague Dependency Treebank. In *Handbook of Linguistic Annotation*, pages 555–594. Springer.

Yusuke Miyao, Stephan Oepen, and Daniel Zeman. 2014. In-house: An ensemble of pre-existing off-the-shelf parsers. In *Proceedings of the Eighth International Workshop on Semantic Evaluation (SemEval 2014)*, pages 335–340, Dublin, Ireland. Dublin City University.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajič, Daniel Hershcovich, Bin Li, Tim O'Gorman, Nianwen Xue, and Daniel Zeman. 2020. MRP 2020: The Second Shared Task on Cross-framework and Cross-Lingual Meaning Representation Parsing. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 1 – 22, Online.

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

Petr Sgall. 1967. Functional sentence perspective in a generative description. *Prague Studies in Mathematical Linguistics*, 2:203–225.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Springer Science & Business Media.