Zero-shot cross-lingual Meaning Representation Transfer: Annotation of Hungarian using the Prague Functional Generative Description

Attila Novák and **Borbála Novák** and **Csilla Novák**
MTA-PPKE Hungarian Language Technology Research Group,
Pázmány Péter Catholic University, Faculty of Information Technology and Bionics
Práter u. 50/a, 1083 Budapest, Hungary
{surname.firstname}@itk.ppke.hu

Abstract

In this paper, we present the results of our experiments concerning the zero-shot cross-lingual performance of the PERIN sentence-to-graph semantic parser. We applied the PTG model trained using the PERIN parser on a 740k-token Czech newspaper corpus to Hungarian. We evaluated the performance of the parser using the official evaluation tool of the MRP 2020 shared task. The gold standard Hungarian annotation was created by manual correction of the output of the parser following the annotation manual of the tectogrammatical level of the Prague Dependency Treebank. An English model trained on a larger one-million-token English newspaper corpus is also available, however, we found that the Czech model performed significantly better on Hungarian input due to the fact that Hungarian is typologically more similar to Czech than to English. We have found that zero-shot transfer of the PTG meaning representation across typologically not-too-distant languages using a neural parser model based on a multilingual contextual language model followed by a manual correction by linguist experts seems to be a viable annotation scenario.

1 Introduction

Two workshops on Cross-Framework Meaning Representation Parsing (MRP) hosted by the 2019 and 2020 editions of the Conference on Computational Natural Language Learning (CoNLL), featured two editions of a shared task, where implementations of parsers turning raw text into different flavors of meaning representation graphs competed with each other.

The first MRP 2019 task (Oepen et al., 2020) involved only English as the object language, and 5 frameworks of meaning representation were featured: DM, PSD, EDS, UCCA and AMR. Two of these frameworks, DM (DELPH-IN MRS Bi-Lexical Dependencies, Ivanova et al., 2012), and

PSD (Prague Semantic Dependencies Hajič et al., 2012; Miyao et al., 2014) are simple bi-lexical dependency graphs generated from core predicate-argument structure of a rich syntactic-semantic annotation based on general theories of grammar. The underlying linguistic theory is Head-Driven Phrase Structure Grammar (HPSG, Pollard and Sag, 1994) with Minimal Recursion Semantics (MRS, Copestake et al., 2005) for DM and Prague Functional Generative Description (FGD, Sgall et al., 1986) for PSD.

In DM and PSD, the nodes are surface word forms. The other three frameworks feature more complex graphs that contain nodes not in one-to-one relation to input words. Elementary Dependency Structures (EDS) are based on English Resource Grammar (Flickinger et al., 2017) aka English Resource Semantics (ERS) (Flickinger et al., 2014) annotation[1] that was turned into a variable-free semantic dependency graph consisting of labeled graph nodes representing logical predications and edges representing labeled argument positions. The conversion from ERS to EDS discards information on semantic scope. The nodes are anchored to spans of the input string.

Universal Conceptual Cognitive Annotation (UCCA, Abend and Rappoport, 2013) is an abstract annotation featuring only purely semantic categories and structure. The foundational layer of UCCA (featured in the shared task) consists of a very basic set of semantic categories like Process, Argument, State, "Adverb" (modifier) etc., that are used as labels on edges linking unlabeled nodes representing semantic units and surface word forms.

Finally, Abstract Meaning Representation (AMR, Banarescu et al., 2013) features graphs comparable to EDS, but with more abstract predication labels due to application of lexical decomposition

---

[1]This is the same HPSG/MRS-based formalism DM is based on

and normalization towards verbal senses e.g., representing 'similar' as the verbal sense 'resemble'. In contrast to EDS (and the rest), AMR nodes are not explicitly anchored to spans of the surface form.[2]

The graphs in the MRP shared tasks were presented in a JSON-lines-based Uniform Graph Interchange Format, and participants were asked to design and train systems that predict sentence-level meaning representations in all frameworks in parallel to foster transfer and multi-task learning.

In the MRP 2020 shared task (Oepen et al., 2020), DM was dropped, as EDS is a richer representation derived from the same resource. PSD was also replaced by a richer meaning representation, Prague Tectogrammatical Graphs (PTG, Zeman and Hajic, 2020), derived from the same Prague Functional Generative Description (FGD), but retaining more of the original Tectogrammatical annotation than PSD.

EDS, UCCA, and AMR were retained, and Discourse Representation Graphs (DRG, Abzianidze et al., 2020) were added as a new meaning representation. DRG is a graph encoding of Discourse Representation Structures (DRS), the meaning representations at the core of Discourse Representation Theory (DRT; Kamp and Reyle, 1993). This model handles many challenging semantic phenomena from quantifiers to presupposition accommodation, and discourse structure.

In addition to English data, the MRP 2020 task covered new languages, one for each of four of the five covered formalisms (except EDS): Czech for PTG, German for UCCA and DRG, and Chinese for AMR.

Training and evaluation data of the MRP shared task was only available to shared task participants distributed to them by the Linguistic Data Consortium (LDC), since part of the data is based on LDC-owned material, the WSJ part of the Penn Treebank (PTB). The shared task site states that, upon completion of each competition, subsets of task data that are copyright-free (including system submissions and evaluation results) will be made available for public, open-source download. However, unfortunately, we have not found a public release of the data.

However, one of the top performing systems at MRP 2020, the PERIN parser (Samuel and Straka, 2020), which was developed at the Institute of

Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, Prague, was made available at the ÚFAL GitHub repo[3] including pretrained models for the PERIN submission to the shared task. There is also a link to an Interactive demo on Google Colab. This facilitated testing the models on various inputs. Positive subjective impressions of the performance of the English and especially the Czech PTG model of the parser on Hungarian input prompted us to perform the experiment described here evaluating the zero-shot cross-lingual performance of the model. Of the models available, we selected PTG, because

- the categories/concepts it operates with looked immediately familiar,

- the annotation it generated seemed reasonable and detailed,

- the non-English model covers Czech, a language sharing many typological features with Hungarian (rich morphology, relatively free word order, pro drop etc.),

- the model was trained on a sizable 740k-token corpus,

- a rather detailed 1255-page annotation manual (Mikulová et al., 2006) of the underlying Prague tectogrammatical annotation is available in English, and

- performance of the parsers (also of PERIN, in particular) on the Czech PTG data reported in the MRP 2020 task results was relatively high.

Concerning the other formalisms featured in the MRP 2020 task, we had the following impressions, further motivating our model selection:

- Annotation in the UCCA foundational layer is rather coarse-grained compared to PDT (a handful of edge label types, no annotation on nodes).[4] In spite of this, all systems participating in MRP 2020 performed relatively poorly ($F_1 < 0.5$ on edge labels) on UCCA. This might indicate consistency problems with the UCCA annotation.

- While reported performance of the best parsers is generally good on DRG, our impression of DRG output generated from our Hungarian test corpus was that it made rela-

---

tively little sense.

- Performance of the best parsers was also good on EDS. However, an EDS-style model is trained only for English. The model struggles on Hungarian input completely misinterpreting important constructions. This seems to be due to typological differences between Hungarian and English. E.g., grammatical relations expressed by prepositions and word order are mainly expressed by suffixes in Hungarian, the latter being an agglutinative language. The EDS model often fails to properly recognize most of these relations (locations, times, possessive constructions, constituents not in canonical positions for English etc.), because suffixes are not independent tokens in Hungarian.[5] There is also pro drop in Hungarian, and this phenomenon affects a high proportion of clauses (see section 4.3.1), but the EDS model fails to recover all such covert pronouns.

The PTG annotation the Czech PERIN model was trained on is derived from the Prague Tectogrammatical Annotation, an elaborate system of deep linguistic analysis based on a many-decade-long tradition of dependency-grammar-based linguistic research. The Prague Dependency Treebank (PDT) and the Prague Czech–English Dependency Treebank (PCEDT; Hajič et al., 2012), from which PTG data was derived, embody an awe-inspiringly immense amount of decade-long annotation work. In addition to the deep syntactic annotation we review here, P(CE)DT annotation includes morphological annotation and a dependency-based shallow 'analytical' syntactic annotation of the underlying text. The tectogrammatical analysis was generated based on these surface-level representations, and then manually checked and corrected.

2 Related Work

Although the MRP 2020 shared task featured a "cross-lingual" track, it only meant in practice that parsers were trained and tested on data in more than one language for meaning representations that had such annotation available. Transfer from one language to another was not tested there.

The Prague dependency annotation scheme has

been ported to languages other than Czech or English, examples including the Slovak Dependency Treebank (Gajdošová et al., 2016), the PAWS Treebank (including Polish and Russian in addition to English and Czech, Nedoluzhko et al., 2018), and the Prague Arabic Dependency Treebank (Hajič and Zemánek, 2004). However, all these syntactic annotations have been created manually.

At the same time, there is a significant body of research literature describing work concerning cross-lingual transfer using deep-neural-network-based models. Multilingual pre-training of contextual language models like multilingual BERT (Devlin et al., 2019) and XLM-RoBERTa (Conneau et al., 2020) facilitated this kind of knowledge transfer. These models have been used to train massively multilingual syntactic dependency parsers (Kondratyuk and Straka, 2019), zero-shot named entity recognizers (Wu et al., 2020) etc, with even specific multilingual benchmarks prepared for testing the cross-lingual generalization capability of models on various tasks such as sentence-pair classification, structured prediction (POS tagging, NER), question answering, natural language inference and sentence retrieval (Hu et al., 2020).

In this paper, we examine a zero-shot approach to meaning representation transfer, which belongs to the structured prediction problem class. Few studies address this topic, because evaluation requires tedious manual work.

3 Method

The approach we took was the following. We had a 150-sentence Hungarian corpus annotated by the PERIN Czech PTG model. We turned this into a gold standard Hungarian annotation by manually correcting the output of the parser following the annotation manual of the tectogrammatical level of the Prague Dependency Treebank (Mikulová et al., 2006). Members of the annotation team had solid training in theoretical and computational linguistics and cognitive science encompassing both dependency syntax and formal semantics. However, we had to understand and learn details of the annotation scheme during the process, which required substantial effort. Fortunately, examples in the Annotation Guidelines have English translation. However, only a few illustrative examples have a full tree representation. We had to reiterate and rediscuss our solutions several times to converge on an annotation that we considered consistent with

---

[5]Note, however, that the English PTG model, which utilizes a rich set of edge label categories to encode grammatical relations, seems to be much less affected by typological differences.

what is described in the PDT annotation guidelines. 50-sentence folds were annotated, discussed and reannotated several times, as our understanding of the annotation scheme evolved during the process. Access to PCEDT would have been very helpful, however, only the Czech part of treebank is available on line,[6] so we could not take a look at the English translations, or efficiently search for specific constructions not being speakers of Czech.

When doing the manual annotation correction, we had to refrain from making modifications to the annotation scheme if we 'disliked' the way a specific phenomenon is handled (or ignored) in the original scheme. We also tried to refrain from interpreting dubious situations 'the way we would have made it', we tried to figure out instead, how ÚFAL experts would do it. We assumed that if the parser more-or-less consistently generates some sort of sufficiently sensible annotation for a specific construction, it reflects a deliberate annotation pattern in the training data.

The serialized JSON lines graph representations output by the parser cannot be edited themselves, although they can be visualized using `mtool`,[7] the Swiss Army Knife for Graph-Based Meaning Representation. We thus created a converter from the JSON lines graph representations to CoNLL-U[8] (using anchors to project the data) and vice versa, and we edited the graphs in the CoNLL-U format. We used `mtool` to visualize our gold standard solutions while we edited the annotations and also to evaluate the zero-shot output of the parser against the edited gold standard version. Based on the graph configurations, `mtool` creates potential node-to-node mappings between the two graphs, so the fact that the ordering of nodes is changed during conversions is not a problem from the point of view of evaluation and visualization.

3.1 The PERIN parser

The name of the parser, PERIN, is motivated by the fact that it embodies a permutation-invariant model that predicts all nodes at once in parallel and

is trained using a permutation invariant loss function that is not sensitive to the ordering of nodes (Samuel and Straka, 2020).

The language model the parser uses as neural input 'features' when inferring the graph annotation based on the input tokens is XLM-RoBERTa (base). XLM-R (Conneau et al., 2020) is the encoder part of a transformer model pretrained originally on 2.5TB of filtered CommonCrawl data in 100 languages including Czech and Hungarian to predict masked word forms. This underlying multilingual neural language model makes an essential contribution to the decent zero-shot cross-lingual performance we encountered possible, enabling the parser to output sensible annotation for input in a language the parser itself was not trained to handle originally.

The model uses relative string encodings to predict node labels that map anchored token strings onto label strings. Specifically, in the PTG model, lemmata ('t-lemmata') are used as node labels. This mechanism performs well (as shown by MRP 2020 evaluation results) when parsing text in the same language the model was trained on. However, in our case, applying Czech lemmatization patterns to Hungarian input unsurprisingly resulted in funny lemmata. Nevertheless, since PTG is a 'Flavor-1' model, i.e. nodes are anchored to spans in the input (practically to tokens), external lemmatization can be used to fix the node labels. Since tokens could be linked to nodes, we could also evaluate the annotation ignoring the ill-formed lemmata.

In contrast, our initial probing of the model indicated that grammatical/semantic relations among content words (edge labels in the graph, 'functors' and 'subfunctors' in PDT terminology) seem to carry over relatively well to Hungarian. And it was this aspect of the annotation that we wanted to concentrate on.

4 Results

In addition to visualization, we also used `mtool` to evaluate the zero-shot output of both the Czech and the English PTG models against the edited gold standard version of the test corpus. English PTG has less node features than the Czech model, and also the edge labels generally contain no subfunctor annotation. The English model also uses different patterns to generate node labels (lemmata), so the performance of the models would not be comparable without applying some sort of normalization

---

PTG-en	P	R	F
tops	1	1	1
labels	0.843	0.763	0.801
properties	0.558	0.222	0.317
anchors	0.864	0.839	0.852
edges	0.682	0.558	0.614
attributes	0.714	0.546	0.619
all	0.734	0.499	0.594
PTG-cz	P	R	F
tops	1	1	1
labels	0.842	**0.857**	**0.849**
properties	0.932	0.943	0.937
anchors	0.844	0.854	0.849
edges	**0.704**	**0.690**	**0.697**
attributes	**0.745**	**0.665**	**0.703**
all	0.849	0.849	0.849

Table 1: Zero-shot performance of the English and Czech PTG models (PTG-en vs. PTG-cz) on the Hungarian test set. Ignore the apparent performance gap in node properties, as these have not been corrected or normalized.

to the annotations before comparison. The normalization we performed included a) replacement of node labels (lemmata) by the sequence of tokens anchored to the node (with the exception of unanchored tokens, which retained their labels), and b) removal of subfunctor annotation from edge labels (except for subtypes of *coref* and *bridging* relations, as these also have subfunctors in the English annotation).

Performance of the normalized output of the models as returned by `mtool` is compared in Table 1. Node properties have not been corrected or normalized (see Section 4.1), so that row can be ignored. But the Czech model definitely performs better at identifying grammatical relations (edges, attributes) and the difference in node label recall reflects mainly its advantage at identifying zero nodes (due to handling of pro drop and richer annotation of argument coreference relations in light verb constructions).

Below we discuss specific details of the performance of the Czech model on Hungarian input.

4.1 Node properties

The PTG graph representations used in the MRP 2020 shared task were created by automatic conversion from tectogrammatical trees in P(CE)DT. English data comes from the Prague Czech-English Dependency Treebank 2.0 (Hajič et al., 2012)

while the source of Czech data was the Prague Dependency Treebank 3.5. (Hajič et al., 2020). Zeman and Hajic (2020) state that '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.' What we see in fact, is that, with the exception of part of speech, and the `tfa` feature (see Section 4.1.2) all we have in terms of node properties ('grammatemes' in the Prague terminology) are the features that were introduced during the upgrade of the treebanks from version 2.0/2.5 to version 3.0. These properties are consequently not described in the PDT 2.0 annotation manual (Mikulová et al., 2006), but their description can be found in Mikulová et al. (2013). The latter source states that these properties were converted from previous annotation by semiautomatic procedures. Moreover, some of the features that remained (*typgroup, diatgram*) have limited or no relevance from our cross-lingual point of view.

At the same time, we really miss the very important grammatical properties that were omitted (number, person, tense, modality, degree etc.). This results in that almost all relevant information is lost in the annotation of e.g., covert pronouns (see Section 4.3.1) or modal auxiliaries (corresponding to *can, must, will* etc.). The latter are not represented in PDT annotation as independent nodes: they only contribute a feature to the node of verb they combine with. This feature, however, is lost in conversion. Technical rather than practical considerations may have played the major role in the selection of the properties kept. Should another conversion of PDT to PTG ever be performed, we would be very happy to see the missing features in the new version. Moreover, the lack of crucial grammatical features in the representation may play an important role in the parser making errors like establishing coreference relations between pronouns and noun phrases of different person/number (e.g., between 'she' and 'I' in the parse of *Eljön, mert szeretem őt.* 'She will come because I love her.' instead of linking 'she' and 'her'.).

Since much of what we would like to see there is not there, and some of what we do have is irrelevant, we have not performed an exhaustive quantitative evaluation of the mapping of node features. Nevertheless, we make some qualitative observa-

tions concerning the performance of the parser with regard to specific node features present in the annotation in the following sections.

4.1.1 Part of speech

Lexical nodes have at least a part-of-speech property, which is termed 'semantic' in PDT terminology, but it is much less semantic than what you would expect. E.g., nominalized verbs are 'semantic' nouns. There are just a few deviations from syntactic part of speech: deadjectival adverbs corresponding to English *-ly* adverbs are tagged 'semantic' adjectives, and numerals as adjectival or nominal quantifiers. Morphological negation is a feature reflected in the part of speech category set that does not apply to Hungarian. Also non-ly adverbs are sometimes tagged as adjectives, but otherwise part of speech is accurately identified by the model.

4.1.2 Topic-focus articulation

The Czech model also contains a feature related to topic-focus articulation (`tfa`). This is an advanced feature rarely found in computational meaning representations. However, topic-focus articulation has been one of the major research directions of the Prague school of linguistics behind PTD, so the presence of a feature like this is not so surprising after all. We would have, however, expected four possible values instead of the actual three: *t* = contextually bound expression (topic), *f* = contextually non-bound expression (new information) , *c* = (contextually bound) contrastive expression. We think that it would be relevant to distinguish contextually non-bound contrastive elements (focus proper) from contextually bound contrastive elements (contrastive topic). We could not determine from the limited description in the annotation manual how specific constructions (e.g., contrasting predicates) should fit into the annotation scheme used in PDT. The parser often assigns values to this feature that seem reasonable, but there are also cases where the annotation is obviously wrong (e.g., assignment of the *f* value to definite expressions). The source of these problems could be among others that word order constraints concerning contrastive elements (focus/contrastive topic) are quite different in Czech and Hungarian (Czech: clause final, Hungarian: preverbal) and that there is no definite article in Czech.

4.1.3 Factual and sentence modality

The model is able to differentiate appeals, requests and questions from assertions, however, quite surprisingly, it often fails to identify potential ('would') and contrafactual ('would have')[9] modalities.

4.2 Identification of grammatical/semantic relations

Although an F score of around 0.7 for edges/attributes (see Table 1) might not seem very great at first sight, this is not bad in fact (especially considering the rich variety of possible labels), and not very much worse than the performance of the same parser model for Czech input (F=0.84/0.78, Samuel and Straka, 2020). The model is especially good at identifying adjuncts (time, place, directional and manner adverbials).

What was a bit disappointing for us is the annotation of predicate argument relations in PDT, which in most cases is limited to two relations called *act* and *pat*. These have nothing to do with real thematic roles like *agent* or *patient*, but are mostly simply placeholders for the first two arguments of any predicate. E.g., the subject of *the window broke* is *act*, while the predicate argument in the valency frame of the copula (i.e., *blue* in *my hat is blue*) is marked as *pat*. But PTG is not alone having uninformative argument labels among the meaning representation schemes in MRP 2020: others have ARG0, ARG1 etc.

4.3 Empty elements

In addition to edge labels, the model also relatively successfully predicts empty elements, such as dropped pronouns and ellipsis as long as similar patterns apply to both the source and the target language.

4.3.1 Pro drop

For example, Czech, similarly to Hungarian, features pro drop: i.e. subject pronouns may optionally be omitted in neutral sentences, as shown in the two side-by-side one-word sentences in (1). In the case of Hungarian, there is a lack of overt subject pronouns in most cases when the pronoun is not emphasized. It is a nice feature of the model that it includes existentially bound optional arguments in the analyses it generates (e.g., *Olvasok.* is interpreted as 'I am reading (something).')

---

[9] In PDT the value 'irreal' is used.

(1) *Olvasok.   Čtu.*
    read.Prs.1Sg read.Prs.1Sg
    'I am reading. = I am reading (something).'

In Hungarian, the same sentence with an overt pronoun has different interpretations depending on the intonation pattern (2).

(2) *Én olvasok.*
    'I am reading.' (neutral, rare)
    'It is me who is reading.' (focus)
    'As for me, I do read.' (contrastive topic)

However, in cases where the same phenomenon (i.e. pro drop) does not apply to certain pronouns in the source language, the model always fails to predict such covert pronouns in the target language. In Hungarian, for example, object pronouns also undergo pro drop. What makes this possible is that verbal morphology encodes not only subject agreement but also the definiteness of the object, as illustrated in (3). If the morphology of verb form implies the presence of a definite object, then the lack of an overt object implies the presence of an object pronoun (4a). In contrast, there is no object pro drop in Czech (4b), thus the model fails to predict covert object pronouns for Hungarian. Instead, we get the same interpretation with an existentially bound object that we get for *Olvasok* (1).

(3) a. *Olvasok   egy könyvet.*
       read.Prs.1Sg a   book.Sg.Acc
       'I am reading a book.'

    b. *Olvasom   a   könyvet.*
       read.Prs.Def.1Sg the book.Sg.Acc
       'I am reading the book.'

(4) a. *Olvasom.*
       read.Prs.Def.1Sg

    b. *Čtu   to.*
       read.Prs.1Sg that.Acc
       'I am reading it.'

4.3.2   Possessive constructions involving pronouns

The same applies to possessive constructions involving personal pronouns. The Czech (or English) version of these constructs involves a possessive

pronoun determiner followed by a noun, optionally modified by adjuncts (5b). In Hungarian (and many similar agglutinating languages), the construct involves possessive suffixes attached to the noun as inflection, and the presence of an overt pronoun is optional, and, again, is mostly limited to cases of emphasis on the pronoun (5a). Since the possessive pronoun is obligatory in Czech (it is the key element of the construction), the parser trained on Czech data always fails to predict empty personal pronouns involved in possessive constructions in Hungarian, too.

(5) a. *az (én) anyám*
       the (I)   mother.Poss.1Sg

    b. *moje   matka*
       my.Fem.Sg.Nom mother.Fem.Sg.Nom
   'my mother'

4.3.3   Ellipsis

The model also performs reasonably well predicting and reconstructing elliptical structures as long as a similar elliptical construction is present in the language the model was trained on. Both Czech and Hungarian feature gapping in the second clause in coordinated clauses. However, in Hungarian (similarly to e.g., Turkish), gapping in the first clause is also a frequently used construction. As shown in Fig. 1, the parser fails to properly recognize the elliptical structure if the gap is in the first clause (not an option in Czech or English). For the given examples, we get a perfect parse only if the gap is in the second clause, and word order in the first clause is SVO (Fig. 1c).

4.3.4   Zero copula

PDT much predates the Universal Dependencies (UD) project, and in contrast to the lexical content head solution to copula constructions there, the copula is the head in PDT/PTG. In Hungarian, there is a zero copula in the default 3rd person singular present indicative case, so we needed to introduce a new zero copula (`#zerocop`) item to accommodate the annotation of zero copula constructions to the scheme applied in PDT and PTG. As it is, the model fails to parse zero copula constructions due to the analysis above and the fact that there is always an overt copula in Czech.

4.4   Coordination/parataxis

In contrast to subordination, coordination (in the PDT annotation manual: parataxis) is problematic
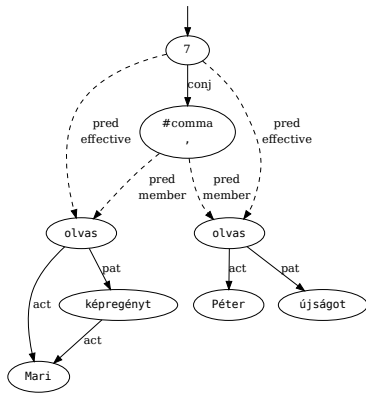
(a) *Péter újságot olvas, Mari képregényt.* 'Peter is reading a newspaper, Mary a comic.' – Gap in second clause, SOV word order in first clause. Minor error in the analysis.
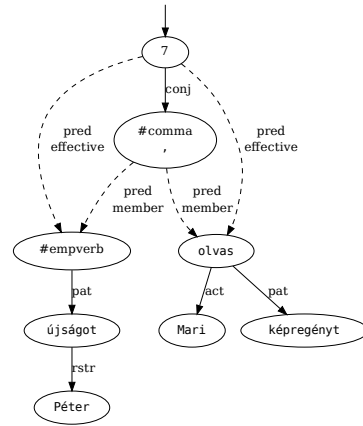
(b) *Péter újságot, Mari képregényt olvas.* – Gap in first clause, SOV word order in second clause. Wrong analysis of first clause.
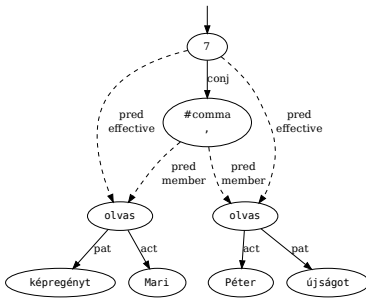
(c) *Péter olvas újságot, Mari képregényt.* – Gap in second clause, SVO word order in first clause. Perfect analysis.
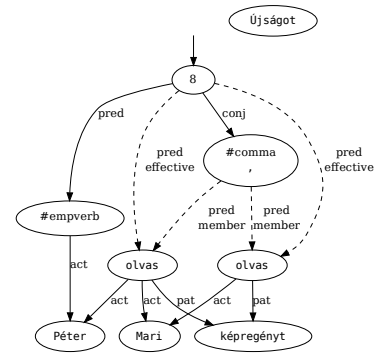
(d) *Újságot Péter, képregényt Mari olvas.* – Gap in first clause, OSV word order in second clause. Wrong analysis.

Figure 1: Output of the Czech PTG grammar for various Hungarian gapping constructions. Only the analysis in 1c is completely correct.

for dependency-based annotation schemes because it is not an endocentric construction. The solution applied in the PTG implementation of PDT structures makes coordinating conjunctions or, in the absence of these, punctuation (commas) the head of coordinate structures, as shown in Fig. 1c. The coordinated predicates there (the two *olvas* 'read' nodes) are *pred–members* of the #comma node (technical head of the coordinate structure), and they are also linked directly by *pred–effective* edges to the node dominating the whole structure, here the root of the sentence graph. The technical head (#comma) is attached using a relation characterizing the paratactic structure (e.g., conjunction, disjunction, apposition etc., here: *conj*) to the node dominating the coordinate structure. This solution is again different from the one applied in UD, how-

ever, it is analogous to the way coordination is represented, e.g., in EDS.

However, two aspects of this solution are problematic. Certain types of coordination express an asymmetric relation like cause, consequence or confrontation, and these types of relations were doubled in the annotation scheme only because they also have a subordinating variant (coordinating *confr, reas* vs. subordinating *contrd, caus*). The distinction is purely syntactic and 99.5% of speakers would have an extremely hard time distinguishing the subordinating variant from the coordinating one. However, the analyses are very different. What is even worse, the representation of the paratactic variant of these constructions completely fails to distinguish which conjunct plays which role, e.g., what is the cause and what is the

8

consequence. These unnecessary syntactic distinctions gave us a hard time during correction of the gold standard data.

Coordinated predicates involving covert subject pronouns were analyzed by the model as verb phrase coordination sharing a single covert subject pronoun rather than assuming two coreferring covert pronouns. We accepted this solution assuming that similar constructions must have been analyzed analogously in the Czech PDT treebank.

4.5 Further problems

The model sometimes fails to integrate parts of the analysis into the whole structure, or, in some cases, completely ignores some part of the input. This often seems to be related to covert elements not attested in the source language like a zero copula or gapping in the first conjunct.

Short function words are sometimes confused with short frequent function words in the source language, and this may result in wrong analysis. E.g., Hungarian *s* 'and' and *a* 'the' are sometimes confused with Czech *s* 'with' and *a* 'and', respectively.

Function words (articles, postpositions, subordinating conjunctions, auxiliaries) are normally merged with content words (the node is anchored on several tokens), but in some cases a partial merge is performed (the function word is anchored both to an independent node of its own and to the node of a content word). This is an error.

The model tokenizes at hyphens, and the hyphen remains unanchored. This is quite problematic for Hungarian, because suffixes (e.g., case endings) are often attached with a hyphen to the stem, and such case endings become independent tokens in the analysis.

5 Conclusion

In this paper we presented our experiment investigating the zero-shot cross-lingual performance of neural parser models based on the PDT/PTG meaning representation formalism. The model yields reasonable performance, and it can be feasibly applied in a semi-automatic annotation scenario. The specific language pairs were Czech–Hungarian vs. English–Hungarian. The former model performs better because the source and the target language share more typological characteristics like rich morphology, free word order, pro drop etc. even though they belong to different language families. Moreover, the PDT/PTG annotation scheme utilizing a rich set of dependency relations as edge labels seems to perform much better than, e.g., EDS where edge labels are completely abstract.

Fortes of the PDT/PTG model:

- some aspects of the model are extremely rich,
- detailed classification and still efficient recognition of adjuncts and covert pronouns, including control, quasi-control, other coreference relations and existentially bound arguments,
- most analyses are easy to interpret.

Points that could be improved:

- important features were discarded in the PDT-to-PTG conversion,
- the *act* and *pat* argument relations are semantically empty (real thematic roles would be very much welcome),
- problems with some asymmetric coordinating structures (unreasonable contrast between e.g., *caus* and *reas*),
- too flat structures (e.g., the attachment of 'rhematizers' to the predicate instead of to what they modify).

Acknowledgements

References

Omri Abend and Ari Rappoport. 2013. UCCA: A semantics-based grammatical annotation scheme. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013) – Long Papers*, pages 1–12, Potsdam, Germany. Association for Computational Linguistics.

Lasha Abzianidze, Johan Bos, and Stephan Oepen. 2020. DRS at MRP 2020: Dressing up discourse representation structures as graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 23–32, Online. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan Sag. 2005. Minimal recursion semantics: An introduction. *Reseach On Language And Computation*, 3:281–332.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dan Flickinger, Emily M. Bender, and Stephan Oepen. 2014. Towards an encyclopedia of compositional semantics: Documenting the interface of the English Resource Grammar. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 875–881, Reykjavik, Iceland. European Language Resources Association (ELRA).

Dan Flickinger, Stephan Oepen, and Emily M. Bender. 2017. *Sustainable Development and Refinement of Complex Linguistic Annotations at Scale*, pages 353–377. Springer Netherlands, Dordrecht.

Katarína Gajdošová, Mária Šimková, and et al. 2016. Slovak dependency treebank. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jan Hajič, Eduard Bejček, Jaroslava Hlavacova, Marie Mikulová, Milan Straka, Jan Štěpánek, and Barbora Štěpánková. 2020. Prague dependency treebank - consolidated 1.0. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5208–5218, Marseille, France. European Language Resources Association.

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 Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3153–3160, Istanbul, Turkey. European Language Resources Association (ELRA).

Jan Hajič and Petr Zemánek. 2004. Prague arabic dependency treebank: Development in data and tools. In *In Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools*, pages 110–117.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom? a contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 2–11, Jeju, Republic of Korea. Association for Computational Linguistics.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, volume 42 of *Studies in Linguistics and Philosophy*. Springer, Dordrecht.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk Žabokrtský. 2006. Annotation on the tectogrammatical level in the prague dependency treebank. annotation manual. Technical Report 30, ÚFAL MFF UK, Prague, Czech Rep.

Marie Mikulová, Eduard Bejček, Jiří Mírovský, Anna Nedoluzhko, Jarmila Panevová, Lucie Poláková, Pavel Straňák, Magda Ševčíková, and Zdeněk Žabokrtský. 2013. From PDT 2.0 to PDT 3.0 (modifications and complements). Technical report, ÚFAL MFF UK, Prague, Czech Rep.

Yusuke Miyao, Stephan Oepen, and Daniel Zeman. 2014. In-house: An ensemble of pre-existing off-the-shelf parsers. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 335–340, Dublin, Ireland. Association for Computational Linguistics.

Anna Nedoluzhko, Michal Novák, and Maciej Ogrodniczuk. 2018. PAWS: A multi-lingual parallel treebank with anaphoric relations. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 68–76, New Orleans, Louisiana. Association for Computational Linguistics.

Stephan Oepen, Omri Abend, Lasha Abzianidze, Johan Bos, Jan Hajic, 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. Association for Computational Linguistics.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago.

David Samuel and Milan Straka. 2020. ÚFAL at MRP 2020: Permutation-invariant semantic parsing in PERIN. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 53–64, Online. Association for Computational Linguistics.

Petr Sgall, Eva Hajičová, and Jarmilla Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. Reidel, Dordrecht.

Qianhui Wu, Zijia Lin, Börje F. Karlsson, Biqing Huang, and Jian-Guang Lou. 2020. Unitrans : Unifying model transfer and data transfer for cross-lingual named entity recognition with unlabeled data. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3926–3932. International Joint Conferences on Artificial Intelligence Organization. Main track.

Daniel Zeman and Jan Hajic. 2020. FGD at MRP 2020: Prague tectogrammatical graphs. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 33–39, Online. Association for Computational Linguistics.