Multi-layer querying in Corpora: Example of Parseme and UD

Bruno Guillaume

Multi-layer querying in Corpora: Example of Parseme and UD

Bruno Guillaume
Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
Bruno.Guillaume@inria.fr

Relevant UniDive working groups: WG1

1 Introduction

The Parseme project (Monti et al., 2018) proposes a large set of annotated data with Verbal Multi-Word Expressions (VMWE). In version 1.2 (Ramisch et al., 2020), 14 languages were covered but with older versions and ongoing work[1], there are data in 26 languages (See Table 3 in appendix for list of languages and the number of sentences for each language).

Most of these data are also released with morpho-syntactic annotations, following the Universal Dependencies (de Marneffe et al., 2021) framework. Some Parseme data are directly annotated on data available in the UD project and then we have both high quality morpho-syntactic annotation and VMWE annotations on the same data. Some other parts of Parseme data, which are not on existing UD data, comes with an automatic morpho-syntactic annotation, built with UDPipe (Straka et al., 2016), thus following the UD annotation framework.

In this abstract, we present an encoding of the two annotation layers in a common structure, using graph encoding of both UD and VMWE annotations. With this encoding, it is possible to use the graph-based tools to work with the data. In this abstract, we use the GREW tool (Guillaume, 2021) to make queries on the two layers.

In Section 2, we explain the encoding and in Section 3, we show how queries can be run and give a few examples.

2 Graph encoding

The two annotation layers are technically stored in a common format (CUPT[2]), but it is not straightforward to consider both in the same structure. In UD, each sentence is split in a sequence of tokens and a annotated VMWE contains several tokens, but not necessarily contiguous (for instance in ***Take*** *a*

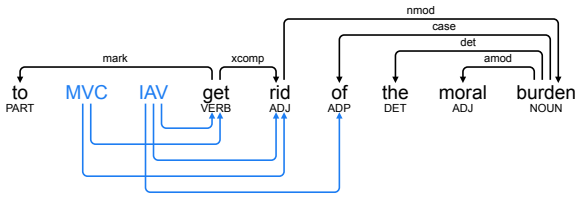

Figure 1: Graph representation (simplified) of two overlapping VMWE annotations

look *!!!*). Moreover, the same token can be implied in more than one VMWE annotation items. In **Parseme-EN** for instance, the same sentence have the two following overlapping annotations: *[...] to* ***get rid of*** *the moral burden [...]* with label **IAV** (Inherently adpositional verbs) and *[...] to* ***get rid*** *of the moral burden [...]* with label **MVC** (Multi-verb constructions).

In order to cover these cases, we propose to represent each sentence as a graph with two kinds of nodes and two kind of edges. In addition to morphosyntactic nodes and edges (drawn in black in Figure 1), for each VMWE annotation a new node is added, labelled with the kind of VMWE and the new node is to each tokens which is part of the VMWE (with a specific type of edges). All VMWE related structure (nodes and edges) is drawn in blue in Figure 1.

3 Multi-layer queries

The benefit of having the two annotation layers in the same structure is that it is possible to make queries which refer to both layers and to make cross observation. To do this, we use the GREW tools which allows to write graph queries that can be executed on the Parseme corpora represented has above.

For querying treebanks, the GREW tool can be used online with GREW-MATCH[3] (on a predefined set of treebanks) on through the GREWPY Python library. With a simple request, and using the clustering feature, we can observe the number of VMWEs

[1] https://gitlab.com/parseme/corpora
[2] http://multiword.sourceforge.net/cupt-format
[3] http://parseme.grew.fr

by type in each language (Table 1 in appendix).

Graph requests can be used to distinguish VMWE annotation which are overlapping with another one with the ones without overlapping. The request below correspond to the "without overlapping" case: lines 1-3 is a request for any VWME (without the tag NotMWE) and lines 4-7 filter out cases where some `MWE2` exists, which shares a token `X` with the one previously found `MWE1`.

```
1 pattern {
2   MWE1 [label<>NotMWE]
3 }
4 without {
5   MWE2 [label<>NotMWE];
6   MWE1 -> X; MWE2 -> X;
7 }
```

For the "with overlapping" case, the request is the same where the keyword `without` is replaced by the keyword `without` (line 4). Table 1 shows the ratio of overlapping VMWE for each language.

3.1 Error mining

Queries can be used to do error mining in the treebanks. We give here two simple queries which can be used on all languages. Of course, as seen in the next subsection, exploration can be done more in depth for given language, through specific queries.

First, we query the different treebanks to search for VMWE containing only one token. Intuitively this should not happen, by definition of multi-word expression. With the request below, we request for such a VMWE:

```
1 pattern {
2   MWE [label];
3   MWE -[parseme=MWE]-> N;
4 }
5 without {
6   MWE -[parseme=MWE]-> X
7 }
```

Secondly, we may wonder if we can have a VMWE which contains neither a VERB nor an AUX in the corresponding tokens. This can be explored with the request:

```
1 pattern {
2   MWE [label];
3 } without {
4   MWE -[parseme=MWE]-> V;
5   V[upos=VERB|AUX]
6 }
```

Table 3 in appendix gives the numbers of occurrences in each language for these two requests. For the first one, there are 4 languages with an high number of occurrences: Hungarian (5735), Chinese (4420), Swedish (1627) and German (1270). All remaining languages have no or few (above 20) occurrences of the request. This shows that the notion of tokenisation is considered differently in both projects. For the second request, the median of the number of occurrences in the 26 treebanks is 92, with 3 treebanks above 1000 occurrences. This shows that the definition what is a "verb" in Parseme is not fully aligned with the UD policy.

3.2 Example: VPC.FULL

In this section, we give a few examples of requests which can be used in GREW-MATCH to explore how some specific class of VWME is annotated in one treebank. The example runs on VPC.FULL and on English data.

First, we can have a look at the distribution of this kind of VMWE according to the number of tokens implied[4]. We observed 368 occurrences[5] of this label, all of them having exactly two tokens.

Another request[6], specifying the two tokens `N1` and `N2`, can display the distribution of the `upos` of the tokens in the following table which shows that two constructions `VERB-ADP` and `VERB-ADV` covers all but 5 cases.

N1.upos \ N2.upos	199 ADP	165 ADV	2 NOUN	2 VERB
363 VERB	199	164		
5 NOUN		1	2	2

Exploring further[7], we observed in the 199 `VERB-ADP` cases, a large majority (185) of annotation where the `VERB` is linked to the `ADP` with relation `compound:prt`. Other cases are: no direct relation between the two nodes (9 cases), relation `advmod` (3 cases), `compound` (2 cases). Similarly[8], we observed in the 164 `VERB-ADV` cases, a majority (93) of annotation where the `VERB` is linked to the `ADP` with relation `compound:prt`. Other cases are: relation `advmod` (66 cases), no direct relation between the two nodes (2 cases), `compound`, `obl` and `xcomp` (1 case fo each).

These irregularities in the annotation would require a careful inspection by a native English speaker but we can already see a bunch of annotation inconsistencies either in the UD annotation layer or in the Parseme one.

[4] http://parseme.grew.fr/?custom=63cfdecd4ed4e

[5] The numbers of this section are based on requests done on 2023/02/02, they may changed when the data is updated. Requests on a stable data from a release will be provided for final version.

[6] http://parseme.grew.fr/?custom=63cfe3d7095fe

[7] http://parseme.grew.fr/?custom=63cfe5673f17f

[8] http://parseme.grew.fr/?custom=63cfe782a12cb

References

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

Bruno Guillaume. 2021. Graph Matching and Graph Rewriting: GREW tools for corpus exploration, maintenance and conversion. In *EACL 2021 - 16th conference of the European Chapter of the Association for Computational Linguistics*, Kiev/Online, Ukraine.

Johanna Monti, Savary Agata, Marie Candito, Verginica Barbu Mititelu, Bejček Eduard, Cap Fabienne, Čéplö Slavomir, Silvio Ricardo Cordeiro, Eryiğit Gülşen, Voula Giouli, Maarten van Gompel, HaCohen-Kerner Yaakov, Kovalevskaitė Jolanta, Krek Simon, Liebeskind Chaya, Carla Parra Escartín, Lonneke van der Plas, Qasemizadeh Behrang, Ramisch Carlos, Federico Sangati, Stoyanova Ivelina, and Vincze Veronika. 2018. Parseme multilingual corpus of verbal multiword expressions.

Carlos Ramisch, Agata Savary, Bruno Guillaume, Jakub Waszczuk, Marie Candito, Ashwini Vaidya, Verginica Barbu Mititelu, Archna Bhatia, Uxoa Iñurrieta, Voula Giouli, Tunga Güngör, Menghan Jiang, Timm Lichte, Chaya Liebeskind, Johanna Monti, Renata Ramisch, Sara Stymne, Abigail Walsh, and Hongzhi Xu. 2020. Edition 1.2 of the PARSEME shared task on semi-supervised identification of verbal multiword expressions. In *Proceedings of the Joint Workshop on Multiword Expressions and Electronic Lexicons*, pages 107–118, online. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

A Example Appendix

Note: At the time of the final version, the last version (1.3) was not released. The tables are built from the data freezed by the Parseme core team on 2023/02/27. The data of the official release 1.3 may be slightly different.

Language	MVC	LS.ICV	LVC.cause	IRV	VID	VPC.full	VPC.semi	IAV	LVC.full
Arabic	303	0	5	2678	0	0	581	0	1182
Bulgarian	0	0	222	3223	1260	0	0	90	1909
Czech	0	0	0	10000	1613	0	0	0	2923
German	0	0	33	322	1437	1744	194	0	311
Greek	51	0	179	1	2841	143	0	0	5293
English	51	0	51	0	187	368	53	71	333
Spanish	713	0	81	714	327	1	0	511	392
Basque	0	0	214	0	880	0	0	0	3152
Farsi	0	0	0	1	17	0	0	0	3435
French	22	0	97	1501	2157	0	0	0	1878
Irish	0	0	118	0	106	28	20	187	200
Hebrew	0	0	223	0	1108	153	0	0	1049
Hindi	306	0	26	0	61	0	0	0	641
Croatian	0	0	147	1193	293	1	0	1388	880
Hungarian	0	0	401	0	104	5156	956	0	1143
Italian	33	37	174	1144	1484	105	2	497	734
Lithuanian	0	0	25	0	308	0	0	0	479
Maltese	2	0	1	1	518	4	0	0	700
Polish	0	0	314	3688	833	0	0	0	2478
Portuguese	18	0	127	1021	1306	0	0	0	3954
Romanian	0	0	182	3826	1644	0	0	3340	516
Slovenian	0	0	64	1626	724	0	0	710	239
Serbian	0	0	69	564	269	0	0	0	402
Turkish	5	0	0	0	4141	0	0	0	3583
Chinese	3826	0	177	0	973	0	4629	0	1214

Table 1: Numbers of occurrences of VMES with theirs labels.

Language	Yes	No
Bulgarian	0.00%	100.00 %
Maltese	0.16%	99.84 %
Turkish	0.57%	99.43 %
Farsi	0.58%	99.42 %
Lithuanian	0.74%	99.26 %
Serbian	1.38%	98.62 %
Slovenian	1.40%	98.60 %
Basque	1.77%	98.23 %
Swedish	2.31%	97.69 %
Hebrew	2.88%	97.12 %
Polish	2.95%	97.05 %
French	3.04%	96.96 %
Chinese	3.38%	96.62 %
Czech	3.78%	96.22 %
Portuguese	4.17%	95.83 %
Arabic	4.27%	95.73 %
English	4.67%	95.33 %
Greek	4.82%	95.18 %
Irish	4.86%	95.14 %
Hungarian	5.54%	94.46 %
German	6.90%	93.10 %
Italian	12.19%	87.81 %
Hindi	12.86%	87.14 %
Romanian	18.46%	81.54 %
Spanish	22.82%	77.18 %
Croatian	28.86%	71.14 %

Table 2: Ratio of VMES which overlap with another annotation (languages sorted by values of the ratio).

Language	# sentences	one_token	no_verb
Arabic	7483	17	1302
Bulgarian	21599	11	416
Czech	49431	0	790
German	8996	1268	126
Greek	26175	1	26
English	7436	4	11
Spanish	5515	2	23
Basque	11158	0	4
Farsi	3617	1	1
French	20961	5	2
Irish	1705	3	214
Hebrew	19200	42	264
Hindi	1684	0	0
Croatian	6133	0	146
Hungarian	6159	5745	5901
Italian	15728	9	65
Lithuanian	11104	0	12
Maltese	10600	13	59
Polish	23547	0	836
Portuguese	32062	1	26
Romanian	56664	0	5
Slovenian	27825	0	0
Serbian	3586	0	91
Swedish	6026	1614	92
Turkish	22306	6	330
Chinese	48929	5382	526

Table 3: Numbers of occurrences of VMES with one token (column `one_token`), of VMES without any verbal token (column `no_verb`).