Annotating Constructions with UD: the experience of the Italian Constructicon

Ludovica Pannitto
University of Bologna
ludovica.pannitto@unibo.it

Lucia Busso
Aston University
l.busso@aston.ac.uk

Giulia Rambelli
University of Bologna
giulia.rambelli4@unibo.it

Beatrice Bernasconi
University of Torino
beatrice.bernasconi@unito.it

Flavio Pisciotta
University of Salerno
fpisciotta@unisa.it

Francesca Masini
University of Bologna
francesca.masini@unibo.it

Relevant UniDive working groups: WG1, WG2

1 Constructicography

In constructionist theory, the constructicon is the inventory and network of constructions (henceforth, *cxns*) makes up the full set of linguistic units in a language. Constructicography (Lyngfelt et al., 2018) then needs to deal with two interconnected aspects, namely:

- the *dictionary of cxns*: a linguistically-motivated repository of entries including what we consider necessary and sufficient information to represent cxns for multiple purposes (language documentation and learning, etc.);
- the identification and annotation of the actual occurrences of cxns (i.e., constructs) in texts.

We argue that it is necessary to tackle them at the same time and cross-linguistically to properly set the path for organic resource development.

Cxns are defined to be conventional form-function pairings at varying levels of complexity and abstraction (Goldberg, 1995, 2006), ranging from morphology to syntax to discourse. As a matter of fact, linguistic resources are mostly available for the morphosyntactic level of analysis and the UD infrastructure represents the *de facto* standard in this area, providing a framework which has been employed to represent the most diverse set of languages (both typologically as well as in terms of language variation - i.e., genre, spoken vs. written, code switching, etc.). At the same time, various initiatives exist to augment the UD proposed annotation to include morphological analysis (Guillaume et al., 2024; Kyjánek et al., 2020), multiword patterns (as in PARSEME, Ramisch et al. 2020), semantic role labeling (as in the Universal PropBank project Jindal et al. 2022). A proposal also exists for adding an annotation layer for construction-like information (Weissweiler et al., 2024).

2 The Italian Constructicon

The formalization we introduce consists of two distinct but interconnected data structures, namely:

- **the graph of cxns** (`GCxn`), a repository of constructions expressed by a set of constraints and organized into a directed graph;
- **a body of annotated examples** incrementally built by annotating the occurrences of a specific cxn (i.e., constructs) in texts.

2.1 GCxn and the `conll-c` format

Within the graph, each cxn is identified by a unique identifier and a human-intelligible (i.e., conventional or easy to grasp) name, and is described through a set of fields stored in `yaml` format.

Most fields are machine-readable, coherently with the aim to define, as much as possible, the properties of a cxn through a UD-compatible formalization. Formally speaking, a cxn is defined as a set of rooted, directed, acyclic, labeled graphs (i.e., dependency trees), where nodes represent constraints on the cxn components (i.e., tokens and sub-token elements) and edges represent constraints on relations among tokens. These constraint can be expressed in what we call `conll-c` format, a newly introduced convention to maximize compatibility with `conll-u`. Cxns formalized in `conll-c` can be translated into a set of `grew` queries and be used to automatically match UD-parsed sentences in order to find cxn instances.

The `conll-c` format mimics `conll-u` with respect to file format (plain UTF-8 text files, containing either tokens, blank lines or cxn-level comments starting with hash). Cxns, just like sentences, consist of one or more word lines. For clarity reasons, we refer to each element of a cxn as a "token", but our notion of token can apply below the word level (i.e., to affixes or roots in the case of morphological constructions). Listing 1 shows an example of `conll-c` annotation of the mirative construc-

```
# cxn-id = 68
# cxn-name = saltare fuori che V
# function = ref:D is found out unexpectedly
# vertical_links =
# horizontal_links = 167

ID  UD.FORM  LEMMA    UPOS              FEATS                    HEAD  DEPREL
A   _        saltare  VERB              Number=Sing|Person=3     0     root
B   fuori    fuori    ADV               _                        A     advmod
C   che      che      SCONJ             _                        D     mark
D   _        _        VERB, ADJ, NOUN   _                        A     csubj

REQUIRED  WITHOUT                    SEM.FEATS  SEM.ROLES     ADJACENCY  IDENTITY
1         CHILDREN:DEPREL=nsubj      _          _             _          _
1         _                          _          _             _          _
1         _                          _          _             _          _
1         _                          _          Eventuality   _          _
```

Listing 1: CoNLL-C annotation for the cxn *salta fuori che* V 'it turns out that V' (lit. 'it jumps out that V'). The columnar format was split in two lines for space reasons.

tion *salta fuori che* V 'it turns out that V' (id = 68).

Cxn-level comments (starting with hash, #), include:

- the name and identifier of the cxn (`# cnx_id` and `# cxn_name`);
- a short natural-language string representing its function (`# cxn_function`); it can include references to specific tokens (in the form of ref:tok_ID) that allow to link the function definition to actual cxn instances;
- a space separated list of cxn ids to which it is vertically linked (parent cxns, `# vertical_links`);
- a space separated list of cxn ids to which it is horizontally linked (sibiling cxns`# horizontal_links`);
- other optional metadata that describe holistic features of the cxn.

Fields express the constraints that constructs need to fulfill in order to be considered instances of the construction. More specifically: Field 1 (`ID`) is compulsory and expresses the token index by means of uppercase alphabetic letter starting at A for each new cxn. For tokens containing sub-word (i.e., morphological) information, `ID` is formed by the letter of the derived token and an integer starting at 1 for each subcomponent of the token (e.g., A*1, A*2, etc.). Fields 2-7 can be left unspecified (_) if no constraint applies, or can be filled with comma separated values compatible with UD notation. More specifically, `FORM` and `LEMMA` can also be expressed by means of regular expressions. In case of subcomponents of the token:

```
# sent_id = 2_Paisa_FP06072024
# source = http://hdl.handle.net/20.500.12124/3 paisa.raw.utf8/7404393 .
# text = [omitted for space reasons]

1   Poi       poi       ADV    4   advmod      _
2   ci        ci        PRON   4   expl        CXN=345:B
3   siamo     essere    AUX    4   aux         _
4   messi     mettere   VERB   0   root        CXN=345:A
5   a         a         ADP    6   mark        CXN=345:C
6   parlare   parlare   VERB   4   xcomp       CXN=345:D
7   ...       ...       PUNCT  9   punct       _
8   e         e         CCONJ  9   cc          _
9   salta     saltare   VERB   4   conj        CXN=68:A
10  fuori     fuori     ADV    9   advmod      CXN=68:B
11  che       che       SCONJ  14  mark        CXN=68:C
12  era       essere    AUX    14  cop         _
13  davvero   davvero   ADV    14  advmod      _
14  Chris     Chris     PROPN  9   ccomp       CXN=68:D
15  Squire    Squire    PROPN  14  flat:name   _
16  !         !         PUNCT  4   punct       _
17  !         !         PUNCT  4   punct       SpaceAfter=No
```

Listing 2: Annotation for the sentence *Poi ci siamo messi a parlare...e salta fuori che era davvero Chris Squire!!* 'Then we got to talking...and it turns out it was really Chris Squire!!!'. For space reasons, only some fields are shown.

- allomorphs are lemmatized as their morpheme (e.g., in Italian all allomorphs of *in-* such as *ir-*, *il-*, *im-*... are lemmatized as *in-*);
- `UPOS` is set as `BMORPH` (*bound morpheme*);
- as a convention, the full word is the `HEAD` of the principal subword element;
- we extend the tagset with the following labels for morphological relations: `root/m` (the stem for derivation, the head for compounding, or the first element in case of coordinate compounds), `der/m` (links derivational affixes to the stem), `case/m` (links complement to head in subordinate compounds), `mod/m` (links attribute to head in attributive compounds), `conj/m` (links constituents in coordinate compounds).

Field 8 (REQUIRED) encodes whether the token has to necessarily appear in every construct (value 1) or is optional (value 0). Field 9 (WITHOUT) stores values that have to be excluded from matching. It is expressed as a pipe separated list of FIELD_NAME=value. A special case is for expressing that no other token can depend on the token in question through a specific relation, in which case we use the syntax CHILDREN=deprel. Fields 10-11 encode constraints on the ontological class of the token (in case of Ns and ADJs, from Open Multilingual WordNet[1]), its Aktionsart (in case of verbs), and its semantic role (adapted from VerbNet[2] and the Unified Verb Index[3]).[4]. Field 12 (ADJACENCY) refers to the linear adjacency of tokens within the sentence. It is filled with either _ for *any* or the ID of the left-adjacent to the one in object. Lastly, field 13 (IDENTITY) encodes constraints on co-indexification (e.g., if token with ID C has to have the same FORM of token with ID A, the value will be FORM=A)

2.2 Annotated Examples

Examples are maintained in conll-u format. In the MISC column, the value CXN=cxn_id:ID is added for relevant tokens. As shown in Listing 2, a single example can be annotated for more than one construction: in this case, both *salta fuori che* V 'it turns out that V' (id = 68) and *mettersi a* V 'begin to V' (id = 345) are annotated. Moreover, metadata is enriched with a # source field which links the example to the original source, in order to automatically update UD-derived annotation and properly split into train, development, and test files.

3 Building the Constructicon

The Constructicon can be populated both manually and semi-automatically. Each cxn is formalized or revised by a linguist that compiles the yaml description, there including also the conll-c formalization. The cxn is then pushed to GCxn where, through CI/CD tools, it will be translated into a series of grew queries which are used to automatically retrieve sentences that might contain instances of the cxn. These (or a subset) are manually revised by a linguist in order to remove false positives. An algorithm will be then run to check graph consistency and update vertical links. Once the position in the graph is set, all sentences associated with the previously existing cxns can be ideally re-analized and enriched in light of information coming from the newly added cxn (i.e., if cxn N is parent of cxn M in the graph and an example has been added for cxn M, than that example can be automatically annotated as an instance of cxn N, too).

[1] https://omwn.org/doc/topics.html

[2] https://verbs.colorado.edu/verbnet/

[3] https://uvi.colorado.edu/

[4] Obviously, not all of these are matchable in a UD-environment, but they still serve the purpose of building a machine-readable version of a cxn. For the same reason, cxns are linked to https://comparative-concepts.github.io/cc-database/.

References

Adele Goldberg. 2006. *Constructions at Work: The Nature of Generalization in Language*. Oxford University Press.

Adele E Goldberg. 1995. *Constructions: A construction grammar approach to argument structure*. University of Chicago Press.

Bruno Guillaume, Kim Gerdes, Kirian Guiller, Sylvain Kahane, and Yixuan Li. 2024. Joint annotation of morphology and syntax in dependency treebanks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 9568–9577.

Ishan Jindal, Alexandre Rademaker, Michał Ulewicz, Ha Linh, Huyen Nguyen, Khoi-Nguyen Tran, Huaiyu Zhu, and Yunyao Li. 2022. Universal proposition bank 2.0. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1700–1711, Marseille, France. European Language Resources Association.

Lukás Kyjánek, Zdenek Zabokrtskỳ, Magda Sevcikova, and Jonás Vidra. 2020. Universal derivations 1.0, a growing collection of harmonised word-formation resources. *Prague Bull. Math. Linguistics*, 115:5–30.

Benjamin Lyngfelt, Lars Borin, Kyoko Ohara, and Tiago Timponi Torrent. 2018. *Constructicography: Constructicon development across languages*, volume 22. John Benjamins Publishing Company.

Carlos Ramisch, Bruno Guillaume, Agata Savary, 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 Stymme, Abigail Walsh, Hongzhi Xu, Emilia Palka-Binkiewicz, Rafael Ehren, Sara Stymne, Matthieu Constant, Caroline Pasquer, Yannick Parmentier, Jean-Yves Antoine, Carola Carlino, Valeria Caruso, Maria Pia Di Buono, Antonio Pascucci, Annalisa Raffone, Anna Riccio, Fed-

erico Sangati, Giulia Speranza, Renata Ramisch, Silvio Ricardo Cordeiro, Helena de Medeiros Caseli, Isaac Miranda, Alexandre Rademaker, Oto Vale, Aline Villavicencio, Gabriela Wick Pedro, Rodrigo Wilkens, Leonardo Zilio, Monica-Mihaela Rizea, Mihaela Ionescu, Mihaela Onofrei, Jia Chen, Xiaomin Ge, Fangyuan Hu, Sha Hu, Minli Li, Siyuan Liu, Zhenzhen Qin, Ruilong Sun, Chenweng Wang, Huangyang Xiao, Peiyi Yan, Tsy Yih, Ke Yu, Songping Yu, Si Zeng, Yongchen Zhang, Yun Zhao, Vassiliki Foufi, Aggeliki Fotopoulou, Stella Markantonatou, Stella Papadelli, Sevasti Louizou, Itziar Aduriz, Ainara Estarrona, Itziar Gonzalez, Antton Gurrutxaga, Larraitz Uria, Ruben Urizar, Jennifer Foster, Teresa Lynn, Hevi Elyovitch, Yaakov HaCohen Kerner, Ruth Malka, Kanishka Jain, Vandana Puri, Shraddha Ratori, Vishakha Shukla, Shubham Srivastava, Gozde Berk, Berna Erden, and Zeynep Yirmibeşoğlu. 2020. Annotated corpora and tools of the PARSEME shared task on semi-supervised identification of verbal multiword expressions (edition 1.2). LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Leonie Weissweiler, Nina Böbel, Kirian Guiller, Santiago Herrera, Wesley Scivetti, Arthur Lorenzi, Nurit Melnik, Archna Bhatia, Hinrich Schütze, Lori Levin, Amir Zeldes, Joakim Nivre, William Croft, and Nathan Schneider. 2024. UCxn: Typologically informed annotation of constructions atop Universal Dependencies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16919–16932, Torino, Italia. ELRA and ICCL.