CONSTRUCTICON ALIGNMENT WORKSHOP 2022

Guidelines for ConstructiCon Alignment via MoCCA

v.1.0

This document presents analytical and technical guidelines for aligning constructions and constructiCons via MoCCA (Model of Comparative concepts for Constructicon Alignment). These guidelines are jointly developed by the Constructicon-building teams participating in the ConstructiCon Alignment Workshop 2022 (henceforth referred to as *the CBT consortium*).

The overall idea of this enterprise is to connect constructions across and within languages using Comparative Concepts (CCs) as a shared base of comparison (Lyngfelt et al. 2022). The CCs provide language-neutral definitions of linguistic properties, and language-particular constructions may be linked to any and all CCs conforming to properties shared by the construction in question. Thereby the construction will also be connected to other constructions linked to the same CC.

The CCs used are of five types: *constructions*, *strategies*, *semantic content*, *information packaging* and *frames*, as described in Croft (2022) and Ruppenhoffer et al. (2016). The first four types are based on language typology and comprise the set of CCs presented by Croft (2022), whereas the fifth type consists of the set of semantic frames defined in the Berkeley FrameNet 1.7 data release. Please note that the language-neutrally defined constructions employed as CCs, henceforth *CC-constructions*, are not to be confused with language-particular constructions, which will here be called *L-constructions*. Names of particular CCs, of any type, are written in boldface; when needed the CC type will be indicated within parentheses after the name.

L-constructions may be linked to one or more CCs, and related L-constructions may share some CCs but differ with respect to others. Thus, the CC links represent *partial correspondences* and should not be confused with equivalence. Also note that the CCs cannot cover all properties of all constructions in all languages. There will always be language-particular idiosyncrasies not covered by this alignment model.

These guidelines are primarily directed towards ConstructiCon Building Teams (CBTs), but may of course be employed by any linguist wishing to connect or compare a particular set of constructions to other constructions via comparative concepts.

1. Analytical guidelines

This section of the guidelines focuses on the procedures and principles for associating Comparative Concepts, including Frames, with constructions and, if applicable, construction elements (CEs).

1.1. Set of Comparative Concepts

To maintain the same base of comparison across different languages, MoCCA is restricted to the set of Comparative Concepts (CCs) provided by the CBT consortium. For the first version, these include the set of CCs provided by Croft (2022) and the frames in the Berkeley FrameNet 1.7 data release.

The CCs, including frames, are presented as a set of related concepts. The Croftian CCs can be related via a subtype_of, implies, part_of or associated_with relation. Frame CCs can be related via the inheritance, subframe, perspective on, precedes, using, causative of, inchoative of, metaphor and see also relations. Thus, the set of CCs to be used is not a list, but a network of concepts. Such a network can be accessed at https://c5.frame.net.br.

Previous experience with the Global FrameNet Shared Annotation task reveals that teams will often find the need to add new CCs to the set or change the existing ones. This is not a choice teams will be allowed to make on the fly, since it would compromise the alignment. Nonetheless, mechanisms are proposed for dealing with cases where teams cannot find a perfectly matching CC for the construction or construction element under analysis, as it will be shown in the section on Reporting Issues.

1.2. General Principles for Associating CCs with Constructions

1.2.1. Analytical Targets

In the linking model being proposed, CCs can be associated with constructions, construction elements or both. The choice of where to make the association is dependent on two factors:

1. Whether the constructiCon in question models constructional constituency in a way that allows for direct association with construction elements (CEs) in the database structure or not. If it does, then CCs should be associated at the most relevant level, as described in section 2 below. CCs for CEs must be manually associated and cannot be automatically derived from the part_of relation. If not, all applicable CCs should be associated at the level of the construction entry.

2. Whether the CC is a general property of the construction or can be traced down to a particular CE (given that association at CE level is possible). It is important to bear in mind that some CCs are already proposed with a set of associations that can be used in such cases. For example, the CCs **relative clause construction (CXN), relative clause head (CXN)** and **relative clause (CXN)**

are applicable to the whole relative clause construction, its head and the dependent clause, respectively.

1.2.2. Generality

When choosing the CC to be associated with a construction or CE, use the most specific one that is applicable. It is important to keep this guideline in mind for two reasons:

1. The CCs in the linking model are organized in a network. Thus, linking an L-construction to a CC also connects it to related CCs of different generality and, indirectly, to associated L-constructions. This feature may somewhat compensate for differences in granularity between L-construction entries in different constructiCons, through a graph structure identifying the closest corresponding target construction. Linking at too high a level of generality, however, may overgenerate and create less accurate connections. Therefore, try to find the most specific CC possible for any given construction or CE.

2. Language-particular constructiCons may also be organized in inheritance networks. When looking at the network of CCs, mind their degree of generality also in reference to the level of generality/specificity of the construction under analysis in your constructicon.

Sometimes this means that the best solution is to link to two or more co-hyponym CCs rather than a single hyperonym. If a supertype CC captures more subtypes than the ones relevant for characterizing the construction under analysis, then, the relevant subtypes should be associated with the construction, instead of the supertype.

1.2.3. Application

Strategies are typically defined with explicit reference to particular CC-constructions but are often straightforwardly applicable to other constructions as well. In such cases, L-constructions conforming to the strategy in question may be linked to it even if they are not instances of the CC-construction being referred to in the definition – *provided that they do not deviate from the definition in any other way*. It is important, however, that such additional associations are documented according to the instructions in Section 3.1.

For example, the **headless strategy** is defined as: “a [strategy] for the [anaphoric-head construction] in which there is no [overt] morpheme that functions as the [head]”. Thus, it is explicitly tied to the **anaphoric-head cxn** but it also applies to other headless structures. This is a typical case where the restriction “for the anaphoric-head construction” may be disregarded without compromising the rest of the definition. If, however, extended application would affect other aspects of the definition, such extension should be avoided.

The CCs are originally introduced in a text book and defined with respect to certain constructions treated in that book. Hence, the more restricted definitions. For the purposes of the linking model, we are interested in more general applicability.

1.2.4. Perspective

Some CCs come in homologous variants, such as a CC-sem and a CC-cxn sharing (more or less) the same definitions, although perspectivized differently. For example, the CCs **reflexive event (*sem*)** and **reflexive cxn** are covered by the same definition, as a meaning and a type of construction associated with this meaning. In the normal case, an L-construction connected to one of these constructions would also be connected to the other.

There may, however, be cases where an L-construction conforms to one of the variant CCs but not the other. For example, if a motion construction does not require a (conventionalized) motion verb, this may be captured linking to **motion event (*sem*)** but not to **motion verb (*cxn*)**, whereas motion constructions that *do* require a motion verb should be linked to both. Thus, the nuances between different CC perspectives may be exploited to indicate contrasts between different constructions, where applicable.

There may also be homologous CCs that are relevant for a given L-construction but not included in the available set of CCs. Reflexive cxns with a reflexive meaning should be linked to both. However, there is no **reflexive (*str*)** CC, which would be useful for constructions using reflexive markers without a reflexive semantics. In such cases, the need to add a homologous CC-str should be documented according to Section 3.1.

1.2.5. Constructional Inheritance

For constructiCons that model construction inheritance, CCs and frames should be associated only once in an inheritance chain, at the adequate level of generality. For example, if a constructicon has a general construction for relative clauses and four other constructions for subtypes of relative clauses, the more general **relative clause construction (CXN)** CC should be associated to the more general language specific construction, while the subtypes—such as **anaphoric head** and **free relative clause constructions (CXN)**—should be assigned to the daughter constructions.

2. Technical guidelines

This section of the guidelines aim at providing constructicon building teams (CBTs) with information on the requirements for implementing the alignment between constructicons. They cover issues concerning the database format, tools and versioning.

2.1. Database format

Considering the need to preserve the autonomy of different CBTs on how to organize and manipulate their data, the construction alignment data should be split into two main files. These files should be elaborated following the “keep it simple” principle, i.e., they should be easy to read, both by humans and computer algorithms, and they should contain only information relevant for the constructiCon alignment.

The first file, referred to as CC **DB File**, comprises a set of Comparative Concepts agreed upon and provided by the “consortium”. Since the CCs are the main features used to align constructions from different projects, this file should be treated as somewhat static. Changes are expected, but should not be drastic or as fast as other data, as they can potentially change the alignment of all Constructicons. The main content of the file is the version of the CC database and the CCs themselves. Each CC must have its Type ID and names indicated (**construction**, **information packaging**, **strategy**, **semantic content**, **frame**), its alphanumeric unique ID, name and definition. A JSON-like schema for this file looks like this:

- CC database version (available and searchable at CxG ORG website)
- List of CCs:
    - CC Type ID
    - CC Type Name
    - CC ID
    - CC Name
    - CC Definition

Having the CCs organized in a list is useful when searching for a CC to be associated with a construction, but it does not represent the relations between CCs. For example, to have a more complete overview, the information of which CC is a subtype of another can be provided in an extended version of the CC DB File. To achieve that, any standardized graph representation format can be used, such as GEXF, GDF, GML and others.

The second main file stores the linking between a Constructicon and the Comparative Concepts. The **Linking DB File** also uniquely identifies the Constructicon among all others and for that reason must contain an alphanumeric identifier, the Constructicon name, its version and a 3-digit ISO code (639-3) of the language. This file must contain an explicit indication of which CC DB version was used for the linking process. The main contents are two lists, one of constructions and another of construction elements (if explicitly modeled in the project). Each item of the former must contain a construction ID, name and description, the ID of a parent construction when it exists and a list of CC IDs. The construction ID does not need to be global, i.e., it should be unique for the Constructicon in question. The list of CC IDs must contain IDs from the CC DB File in the version specified in the linking file. The required data for construction elements mirrors the ones just described: element ID, name and description, parent element ID and a list of CC IDs. The only difference is that an extra field, construction ID, is required to link the element to its construction. A representation of this schema in a JSON-like format looks like this (a **?** indicates an optional field):

- My DB ID
- My DB Name
- My DB version
- Language ID (ISO)
- CC database version
- Comment ?
- Construction List:
    - Construction ID
    - Construction Name
    - Construction Name (en) ?

- Construction Description
- Construction Description (en) ?
- List of Examples [0-3]
- Parent Construction ID ?
- List of CC IDs
- Comment ?
- Element List ?:
    - Element ID
    - Element Name
    - Element Name (en) ?
    - Element Description
    - Element Description (en) ?
    - Parent Element ID ?
    - List of CC IDs
    - Comment ?

2.2. Tools

Relating constructions to Comparative Concepts is the only way in which data from different projects can be connected. To make this process easier, faster and inconsistency-free, the linking tool can be used. Teams are only required to transform their existing Constructicon data into the same format of the Linking DB File, with the exception of the CC IDs lists.

After the Constructicon is loaded, along with a specific version of the CC database, users can easily link their constructions and elements to CCs via the UI. This tool can be further extended to export the data in different formats and include additional information that is not required by the data schemas described earlier.

Another useful feature for this tool or a separate one is an intuitive visualization of the alignments of different Constructicons (via Linking DB files). This could be a simple graph visualization of a pre-computed alignment or of a subgraph that is interesting for the user. This subgraph could be defined based on the constructions, CEs, CC types or more dynamic filters, such as restricting visualization to neighbors within a certain distance on the graph. Another form of visualization could use flow diagrams (such as Sankey) to better understand the nuances of the alignments between constructions. Naturally, the metrics and algorithms used in this type of visualization must be explicitly defined and chosen after a more complete discussion of their adequacy.

2.3. Versioning

To increase compatibility and preserve the ability of projects to work at their own pace, all of the databases and tools discussed in the previous items need to be versioned. Every database or tool built based on the CC or the Linking databases needs to explicitly include the version of those files that was used as part of the metadata. In the case of the Linking DB Files, the Constructicon projects are expected to update their version according to the changes made. When changing the CC database version used by a linking file, documentation will be provided to guide the automatic or manual update to a new version, depending on the changes made to the CCs by the “consortium”.

3. Reporting issues

This final section presents guidelines for what to do when the previous guidelines fail. In such cases, teams should report an issue. This reporting can be done at the C5 interface (https://c5.frame.net.br)

3.1. Analytical Issues

There are two cases for reporting a missing CC: (1) if there is already an approximate CC in the network, or (2) if there is no such CC.

1. Reporting the lack of best-fit CCs for the Constructions or CEs under analysis when there is an approximate CC available:
    a. Choose the most approximate CC in the network.
    b. Document your choice by indicating the most approximate CC in the network and by suggesting the name for the new best-fit CC and its position in the network. The position must be described using possible relations (subtype_of and associated_to) to an existing CC.
    c. Submit for analysis by the CBT consortium.
2. Reporting the lack of best-fit CCs for the Constructions or CE under analysis when there is NO approximate CC available:
    a. Suggest the name for the best-fit CC and its position in the network. The position must be described using possible relations (subtype_of and associated_to) to an existing CC.
    b. Submit a case for analysis by the CBT consortium.