CorefUD 1.0 File Format Description

The main building blocks in the target representation are **mentions** and **clusters**. A mention in our scheme is a set of words in the sense of UD, that is, nodes in the dependency structure, including empty nodes – zeros. Mentions spanning multiple sentences are supported, too. A mention is specified by its span, i.e., the nodes it contains. Spans of two different mentions can overlap but they cannot be identical. While a typical mention is a contiguous span of the surface text, this is not a requirement and discontinuous mentions are allowed. Analogously, from the perspective of the dependency structure, a typical mention is a connected component of a dependency tree (*catena*), yet we do not require this to be the case, and for automatically parsed corpora we expect recurring violations of this expectation.

Every mention is a member of one (and only one) cluster; the cluster contains all mentions referring to the same **entity** (incl. events). Singletons are clusters that contain only one mention. The entity/cluster ID is thus a required attribute of each mention, besides the mention's span.[1] Mentions have additional attributes, some of which pertain to the whole cluster.

File Format

Our main objective is maximum compliance with the current UD standards. We avoid decisions that would prevent our data from becoming part of a regular UD release.[2]

We adhere to the specification of the **CoNLL-U format**[3] (as opposed to the CoNLL-U Plus extension,[4] which would allow for extra columns for the coreference-related attributes, but unfortunately would disqualify the data from UD releases). We make sure that the harmonized data pass the official UD validation at level 2 (passing the higher levels may not be possible with automatically predicted POS tags and dependency relations).[5]

From the perspective of the CoNLL-U format, coreference is additional annotation that belongs to the MISC column (column 10). While we deliberately avoid the CoNLL-U Plus file format, we argue that this option is very close to it, and users who prefer additional columns for coreference annotation can easily extract the coreference-related attributes from MISC and place them in separate columns, using tabs instead of the MISC column's pipe separators.

The main attribute that we add to the MISC column is called Entity and it identifies all mentions that begin or end at the current word. In the value of the attribute, each mention has an opening or closing bracket, accompanied by the entity/cluster ID. Additional mention attributes are specified at the opening bracket, i.e., at the first word of the mention. For example, Entity=(e8-place-1)e9) means that the current word is the entire span of one mention of entity e8, the corresponding entity type is a place, and the first (and only) word

[1] Cluster IDs are unique across one corpus within CorefUD. For example, `e1` refers to the same cluster everywhere in Czech-PDT but it is not related to `e1` in Czech-PCEDT. This is mainly to prevent confusion when interpreting the data. For coreference purposes it would be sufficient to make the IDs unique within one document, if the corpus has internal document boundaries.

[2] Note however that UD has additional requirements, which only some of our datasets comply with. Most notably, a UD-released treebank must have manually checked POS tags and dependency relations; in most of our datasets, this kind of annotation has been assigned automatically.

[3] `https://universaldependencies.org/format.html`

[4] `https://universaldependencies.org/ext-format.html`

[5] `https://universaldependencies.org/validation-rules.html#levels-of-validity`

```
# global.Entity = eid-etype-head-minspan-infstat-link-identity
# sent_id = GUM_academic_art-3
# text = Claire Bailey-Ross xxx@port.ac.uk University of Portsmouth,
       United Kingdom
1   Claire      Claire      PROPN  NNP  Number=Sing  0  root   0:root
    Entity=(e5-person-1-1,2,4-new-coref|Discourse=attribution:3->57:7
2   Bailey      Bailey      PROPN  NNP  Number=Sing  1  flat   1:flat
    SpaceAfter=No|XML=<w>
3   -           -           PUNCT  HYPH _            4  punct  4:punct
    SpaceAfter=No
4   Ross        Ross        PROPN  NNP  Number=Sing  2  flat   2:flat
    Entity=e5)|XML=</w>
5   xxx@port.ac.uk xxx@...  PROPN  NNP  Number=Sing  1  list   1:list
    Entity=(e6-abstract-1-1-new-sgl)
6   University  University  PROPN  NNP  Number=Sing  1  list   1:list
    Entity=(e7-organization-1-3,5,6-new-sgl-University_of_Portsmouth
7   of          of          ADP    IN   _            8  case   8:case    _
8   Portsmouth  Portsmouth  PROPN  NNP  Number=Sing  6  nmod   6:nmod:of
    Entity=(e8-place-1-3,4-new-sgl-Portsmouth|SpaceAfter=No
9   ,           ,           PUNCT  ,    _            11 punct  11:punct  _
10  United      unite       VERB   NNP  Tense=Past|... 11 amod 11:amod
    Entity=(e9-place-2-1,2-new-coref-United_Kingdom
11  Kingdom     Kingdom     PROPN  NNP  Number=Sing  1  list   1:list
    Entity=e9)e8)e7)
```

Figure 1: Example of the CoNLL-U encoding of English-GUM in CorefUD.

of the mention is also its syntactic head; furthermore, the attribute says that this is the last word of a larger mention belonging to cluster e9, which started at one of the previous words.

In case of a discontinuous mention, each part has its number and the total number of parts in square brackets after the cluster ID: Entity=(e10[1/2] … Entity=e10[1/2]) … Entity=(e10[2/2] … Entity=e10[2/2]).

For an example of the CoNLL-U representation see Figure 1.

Zeros. Universal Dependencies provide a mechanism for inserting **empty nodes** (which may or may not have lexical values assigned to them) in the enhanced dependency graph. We use the empty nodes to represent reconstructed zeros.

Singletons. Both singletons and non-singletons are treated as clusters; a singleton cluster contains just a single mention. As a result, there are substantially more unique cluster IDs for the annotation projects that include annotation of singletons. In future versions, we may add singletons to datasets which did not have them originally, using the UD annotations and/or entity recognition tools.

Bridging. In the current version, bridging relations are understood very broadly as all relations annotated in the source schemes that cannot be considered types of identity coreference. To record bridging relations, we use the MISC attribute Bridge. It connects identity clusters, where one cluster may be part of more than one bridging relation. For example, Bridge=e173<e188:subset,e174<e188:part says that cluster e188 is related to cluster e173 with the subset bridging relation, and to cluster e174 with the part-whole bridging relation. The annotation appears at the first word of a selected mention of cluster e188; it is not repeated at the other mentions of that cluster.

Split antecedents. The MISC attribute SplitAnte points from a cluster to two or more other clusters. For example, SplitAnte=e5<e61,e10<e61 means that cluster e61 anaphorically refers to clusters e5 and e10. The attribute is a property of clusters, saying that the entity with a given cluster ID is equivalent to the union of the smaller entities whose IDs are listed in the value of the attribute. The annotation appears at the first word of a selected mention of cluster e61; it is not repeated at the other mentions of that cluster.

Attributes of clusters and mentions. There are three "standardized" attributes: eid (entity/cluster ID), etype (entity type) and head (index of the head word), stored as a hyphen-separated list. Other attributes may follow. In CorefUD version 1.0, we just copy these additional attributes from the original annotation schemes. In future versions, we anticipate adding a number of modifications to unify the data further, for example the distinction between specific and generic NPs.