**CoNLL-U Format**

We use a revised version of [the CoNLL-X format](http://anthology.aclweb.org/W/W06/W06-2920.pdf) called CoNLL-U. Annotations are encoded in plain text files (UTF-8, [normalized to NFC](http://unicode.org/reports/tr15/), using only the LF character as line break, including an LF character at the end of file) with three types of lines:

1. Word lines containing the annotation of a word/token in 10 fields separated by single tab characters; see below.
2. Blank lines marking sentence boundaries.
3. Comment lines starting with hash (#).

Sentences consist of one or more word lines, and word lines contain the following fields:

1. ID: Word index, integer starting at 1 for each new sentence; may be a range for multiword tokens; may be a decimal number for empty nodes (decimal numbers can be lower than 1 but must be greater than 0).
2. FORM: Word form or punctuation symbol.
3. LEMMA: Lemma or stem of word form.
4. UPOS: [Universal part-of-speech tag](https://universaldependencies.org/u/pos/index.html).
5. XPOS: Language-specific part-of-speech tag; underscore if not available.
6. FEATS: List of morphological features from the [universal feature inventory](https://universaldependencies.org/u/feat/index.html) or from a defined [language-specific extension](https://universaldependencies.org/ext-feat-index.html); underscore if not available.
7. HEAD: Head of the current word, which is either a value of ID or zero (0).
8. DEPREL: [Universal dependency relation](https://universaldependencies.org/u/dep/index.html) to the HEAD ([root](https://universaldependencies.org/u/dep/root.html) iff HEAD = 0) or a defined language-specific subtype of one.
9. DEPS: Enhanced dependency graph in the form of a list of head-deprel pairs.
10. MISC: Any other annotation.

The fields DEPS and MISC replace the obsolete fields PHEAD and PDEPREL of the CoNLL-X format. In addition, we have modified the usage of the ID, FORM, LEMMA, XPOS, FEATS and HEAD fields as explained below.

The fields must additionally meet the following constraints:

* Fields must not be empty.
* Fields other than FORM, LEMMA, and MISC must not contain space characters.
* Underscore (\_) is used to denote unspecified values in all fields except ID. Note that no format-level distinction is made for the rare cases where the FORM or LEMMA is the literal underscore – processing in such cases is application-dependent. Further, in UD treebanks the UPOS, HEAD, and DEPREL columns are not allowed to be left unspecified except in multiword tokens, where all must be unspecified, and empty nodes, where UPOS is optional and HEAD and DEPREL must be unspecified.

See the [tools page](https://universaldependencies.org/tools.html) for a list of tools that work with the CoNLL-U format. There is also a simple JavaScript-based [CoNLL-U file viewer](https://universaldependencies.org/conllu_viewer.html).

**Words, Tokens and Empty Nodes**

To accommodate the representation of both words and (multiword) tokens (see [Tokenization and word segmentation](https://universaldependencies.org/u/overview/tokenization.html)), we adopt an extension of the original CoNLL-X token indexing scheme, where words are indexed with integers 1, 2, 3, …, while (multiword) tokens are indexed with integer ranges like 1-2 or 3-5. Lines representing such tokens are inserted before the first word in the range. These ranges must be nonempty and must not overlap. They have a FORM value – the string that occurs in the sentence – but have an underscore in all the remaining fields except MISC (because the token represents multiple words, each with its own lemma, part-of-speech tag, syntactic head, and so on). This is illustrated in the following annotation snippet, showing only the first three fields for the Spanish sentence *vámonos al mar* (let’s go to the sea):

1-2 vámonos \_

1 vamos ir

2 nos nosotros

3-4 al \_

3 a a

4 el el

5 mar mar

We extract the word sequence by skipping all range IDs:

1 vamos ir

2 nos nosotros

3 a a

4 el el

5 mar mar

We extract the raw token sequence by skipping all integer IDs that are included in a preceding range ID:

1-2 vámonos \_

3-4 al \_

5 mar mar

To accommodate the use of empty nodes for the analysis of ellipsis in the enhanced dependency representation, we adopt a further extension of the indexing scheme from v2. It is possible to insert one or more empty nodes indexed *i*.1, *i*.2, etc. immediately after a word with index *i*(where *i* = 0 for sentence-initial empty nodes). Note that the the numbers after the decimal point must form a sequence starting at 1, i.e. it is not allowed to skip *i*.1 and use *i*.2. In the unlikely case that there are ten or more empty nodes between two real tokens, *i*.10 comes after *i*.9, that is, the entire ID is not ordered numerically as a decimal number. Here is an example showing the use of an empty node in the analysis of the sentence *Sue likes coffee and Bill tea*:

1 Sue Sue

2 likes like

3 coffee coffee

4 and and

5 Bill Bill

5.1 likes like

6 tea tea

Empty nodes must have non-empty ID and DEPS fields and empty fields (i.e. underscores) for HEAD and DEPREL, because they are only part of the enhanced dependency graph. All other fields may contain either underscores or values: for example, they can optionally have a word form and lemma as in the example above.

**Morphological Annotation**

The UPOS field contains a part-of-speech tag from the [universal POS tag](https://universaldependencies.org/u/pos/index.html) set, while the XPOS optionally contains a language-specific part-of-speech tag, normally from a traditional, more fine-grained tagset. If the XPOS field is used, the treebank-specific documentation should define a mapping from XPOS to UPOS values (which may be context-sensitive and refer to other fields as well). If no language-specific tags are available, the XPOS field should contain an underscore for all words.

The FEATS field contains a list of morphological features, with vertical bar (|) as list separator and with underscore to represent the empty list. All features should be represented as attribute-value pairs, with an equals sign (=) separating the attribute from the value. In addition, features should as far as possible be selected from the [universal feature inventory](https://universaldependencies.org/u/feat/index.html) and be sorted alphabetically by attribute names. It is possible to declare that a feature has two or more values for a given word: Case=Acc,Dat. In this case, the values are sorted alphabetically. In sorting, uppercase letters are considered identical to their lowercase counterparts. Feature names must have the form [A-Z][A-Za-z0-9]\*(\[[a-z0-9]+\])? and feature values must have the form [A-Z0-9][A-Za-z0-9]\*.

Here is an example, showing only the first five fields for the Swedish sentence *Då var han elva år* (Then he was eleven years old):

1 Då då ADV AB \_

2 var vara VERB VB.PRET.ACT Tense=Past|Voice=Act

3 han han PRON PN.UTR.SIN.DEF.NOM Case=Nom|Definite=Def|Gender=Com|Number=Sing

4 elva elva NUM RG.NOM Case=Nom|NumType=Card

5 år år NOUN NN.NEU.PLU.IND.NOM Case=Nom|Definite=Ind|Gender=Neut|Number=Plur

6 . . PUNCT DL.MAD \_

Morphological annotation is only provided for words. Tokens that are not words have an underscore in the UPOS, XPOS and FEATS fields.

**Syntactic Annotation**

The HEAD and DEPREL fields are used to encode a dependency tree over words. The DEPREL value should be a [universal dependency relation](https://universaldependencies.org/u/dep/index.html)or a language-specific subtype of such a relation (defined in the language-specific documentation). As in the case of morphology, syntactic annotation is only provided for words, and tokens that are not words have an underscore in both the HEAD and DEPREL fields.

The HEAD and DEPREL values define the basic dependencies which must be strictly a tree. However, in addition to these basic dependencies, treebanks may optionally provide an enhanced dependency representation that specifies additional dependency relations, for example, when dependencies propagate over coordinate structures. The enhanced dependency representation, which in general is a graph and not a tree, is specified in the DEPS field, using a list of head-relation pairs. We use colon (:) to separate the head and relation and (as usual) vertical bar (|) to separate list items and underscore for the empty list. The list is to be sorted by the index of the head: 4:nsubj|11:nsubj.

Note that the DEPS field should be a self-contained representation of the enhanced dependency graph, which means that dependencies that are shared between the basic and the enhanced dependency representations must be repeated in the DEPS field. Here is an example, showing the first nine fields for the English sentence *They buy and sell books*:

1 They they PRON PRP Case=Nom|Number=Plur 2 nsubj 2:nsubj|4:nsubj

2 buy buy VERB VBP Number=Plur|Person=3|Tense=Pres 0 root 0:root

3 and and CONJ CC \_ 4 cc 4:cc

4 sell sell VERB VBP Number=Plur|Person=3|Tense=Pres 2 conj 0:root|2:conj

5 books book NOUN NNS Number=Plur 2 obj 2:obj|4:obj

6 . . PUNCT . \_ 2 punct 2:punct

The dependency relations in DEPREL must have the form [a-z]+(:[a-z]+)?. Certain extensions of this basic format will be allowed in DEPS but not in DEPREL; see the guidelines for the [enhanced representation](https://universaldependencies.org/u/overview/enhanced-syntax.html) for details.

**Miscellaneous**

The final MISC field is for storing any additional information that does not fit into any of the other fields, such as language-specific annotation, any information about other linguistic levels such as discourse, or projective heads and dependency relations (cf. the old PHEAD and PDEPREL fields of the CoNLL-X format). The exact format used in this field should be specified in the treebank-specific documentation, but it has to be formatted as a list that can be split on the vertical bar character (|). If this character is needed in a value within the list, it must be escaped in a way that does not involve the character itself (for example, preceding the character with a backslash will not work and the vertical bar will still be considered a list separator). If the MISC field is not used, it should contain an underscore.

**Untokenized Text**

To facilitate reconstruction of original (pre-tokenization) text, the information on original word segmentation should be kept if available. If it is not available, UD treebanks since release 2.0 must approximate it using detokenization heuristics (see also the sentence-level attribute textbelow).

Every token after which there was no space in the original text should contain SpaceAfter=No in its MISC field. Note that this feature applies to the token level, not to the word level. Syntactic words that are just part of surface tokens will be ignored during detokenization and thus do not need the feature. In the example below, the line indexed 3 does not contain the SpaceAfter feature even though there was no space between *für* and *das* in the underlying sentence. However, if there were no space between *fürs* and the following token, the 3-4 line would have SpaceAfter=No.

Note that columns 5 to 9 are collapsed in the following example.

# text = Er arbeitet fürs FBI (deutsch etwa: „Bundesamt für Ermittlung“).

# text\_en = He works for the FBI (German approx: “Bundesamt für Ermittlung”).

1 Er er PRON … \_

2 arbeitet arbeiten VERB … \_

3-4 fürs \_ \_ … \_

3 für für ADP … \_

4 das der DET … \_

5 FBI FBI PROPN … \_

6 ( ( PUNCT … SpaceAfter=No

7 deutsch deutsch ADV … \_

8 etwa etwa ADV … SpaceAfter=No

9 : : PUNCT … \_

10 „ „ PUNCT … SpaceAfter=No

11 Bundesamt Bundesamt NOUN … \_

12 für für ADP … \_

13 Ermittlung Ermittlung NOUN … SpaceAfter=No

14 “ “ PUNCT … SpaceAfter=No

15 ) ) PUNCT … SpaceAfter=No

16 . . PUNCT … \_

**Other Miscellaneous Attributes**

Besides SpaceAfter=No, there are some other token- or word-level attributes that may be useful in multiple treebanks. These are not required; but if this sort of information is available, it is desirable that it is encoded the same way in all treebanks.

* Translit … transliteration or transcription of the word form to another writing system. Typically this attribute is used in languages that do not write using the Latin script, and the attribute provides some standard romanization.
* LTranslit … analogy of Translit for lemmas.
* Gloss … approximate translation of the word form or the lemma to another language (typically English). If the translation consists of multiple words, they are connected using a hyphen.
* MSeg … morphemic segmentation as commonly used in interlinear glossed text in linguistic literature: a hyphen (“-”) denotes boundary between morphemes, “=” is placed between a clitic and its host word.
* MGloss … glossing of individual morphemes as commonly used in interlinear glossed text in linguistic literature. Hypens and equals-to symbols delimit morphemes as in MSeg, and there should be the same number of morphemes as in MSeg (if MSeg is missing, a single morpheme is assumed). A gloss is either a lexical meaning in English, or a grammatical tag; if multiple words/tags are needed in the gloss of one morpheme, they are joined by a period (“.”). There are no guidelines for the tags ([Leipzig glossing rules](https://www.eva.mpg.de/lingua/resources/glossing-rules.php) are a source of tags that are commonly used). However, most of the tags should probably have a corresponding feature in the FEATS column, and there it must follow the UD guidelines.

**Sentence Boundaries and Comments**

There must be exactly one blank line *after* every sentence, including the last sentence in the file. Empty sentences are not allowed.

Lines starting with the # character and preceding a sentence are considered as carrying comments or metadata relevant to the following sentence. These lines are an integral part of the format as they give the ability to embed metadata together with the sentences. Consequently, any tools compatible with the CoNLL-U format should carry these lines over into their output (unless specifically designed to process them in some way). Comment and metadata lines inside sentences (i.e., between the token lines) are disallowed.

The contents of the comments and metadata is basically unrestricted and will vary depending on the application, but from v2 the following two comments are compulsory for every sentence (and there must be just one comment of each kind per sentence):

* A treebank-wide unique sentence id (sent\_id), formatted as in the examples below. It is assumed that the actual identifier does not contain whitespace characters (while the comment line may contain whitespace around the sent\_id keyword and the equals-to sign). In sentence ids, the slash character (“/”) is reserved for specialized downstream use and should be avoided in UD treebanks. (The specialized use deals with multiple annotations of one sentence within one file, or with parallel data within one file. See [Issue 321](https://github.com/UniversalDependencies/docs/issues/321) for more details. UD releases include some parallel treebanks but these are distributed separately by languages, hence sentence ids with slashes are not used.)
* Comments used to specify the unannotated sentence as a single string (text) should also be formatted as below. If the original text is not available, the providers of the UD treebanks must approximate the text attribute using detokenization heuristics.
  + If you provided the Translit attribute in MISC (see above), maybe you want to also provide the transliteration of the entire sentence as a sentence-level comment; use # translit = .... In contrast to the text attribute, translit is optional.
  + Whether or not you provided the Gloss attribute in MISC (see above), you may want to provide the fluent translation of the sentence to English or another language. Use # text\_en = ... for English (and a similar comment with the corresponding language code for other languages). In contrast to the text attribute, text\_en is optional.

Example:

# sent\_id = 1

# text = They buy and sell books.

1 They they PRON PRP Case=Nom|Number=Plur 2 nsubj 2:nsubj|4:nsubj \_

2 buy buy VERB VBP Number=Plur|Person=3|Tense=Pres 0 root 0:root \_

3 and and CONJ CC \_ 4 cc 4:cc \_

4 sell sell VERB VBP Number=Plur|Person=3|Tense=Pres 2 conj 0:root|2:conj \_

5 books book NOUN NNS Number=Plur 2 obj 2:obj|4:obj SpaceAfter=No

6 . . PUNCT . \_ 2 punct 2:punct \_

# sent\_id = 2

# text = I have no clue.

1 I I PRON PRP Case=Nom|Number=Sing|Person=1 2 nsubj \_ \_

2 have have VERB VBP Number=Sing|Person=1|Tense=Pres 0 root \_ \_

3 no no DET DT PronType=Neg 4 det \_ \_

4 clue clue NOUN NN Number=Sing 2 obj \_ SpaceAfter=No

5 . . PUNCT . \_ 2 punct \_ \_

# sent\_id = panc0.s4

# text = तत् यथानुश्रूयते।

# translit = tat yathānuśrūyate.

# text\_fr = Voilà ce qui nous est parvenu par la tradition orale.

# text\_en = This is what is heard.

1 तत् तद् DET \_ Case=Nom|…|PronType=Dem 3 nsubj \_ Translit=tat|LTranslit=tad|Gloss=it

2-3 यथानुश्रूयते \_ \_ \_ \_ \_ \_ \_ SpaceAfter=No

2 यथा यथा ADV \_ PronType=Rel 3 advmod \_ Translit=yathā|LTranslit=yathā|Gloss=how

3 अनुश्रूयते अनु-श्रु VERB \_ Mood=Ind|…|Voice=Pass 0 root \_ Translit=anuśrūyate|LTranslit=anu-śru|Gloss=it-is-heard

4 । । PUNCT \_ \_ 3 punct \_ Translit=.|LTranslit=.|Gloss=.

**Paragraph and Document Boundaries**

In addition, we define sentence-level and token-level comments (attributes) that mark paragraph and document boundaries. This kind of information is optional and sometimes it is not available (original text is lost, sentences have been shuffled etc.) but if it is available, it should be encoded in a unified way. Document and paragraph boundaries can be useful for various applications, including but not limited to sentence segmentation.

Note that while document boundaries always occur between sentences, paragraph boundaries may under certain circumstances occur in the middle of a sentence (bulleted list items, verse etc.) Document and/or paragraph boundaries are encoded as follows:

* The first sentence of a new document contains a comment that says # newdoc, which can be optionally followed by a document id (newdoc id = wsj2012-01-05). It is not necessary that the first sentence of a CoNLL-U file has the newdoc comment (e.g. if the document is split between development and test data).
* When a paragraph starts at sentence boundary, the first sentence of the paragraph contains a comment that says # newpar, which can be optionally followed by a paragraph id (newpar id = wsj2012-01-05-p1).
* When a new paragraph starts between two tokens of a sentence, the first token of the new paragraph contains the attribute NewPar=Yes in the MISC column. If it is a multi-word token, the attribute will appear in the line of the multi-word token, not in the line of its first syntactic word.

Note that the annotation defined in this section is observed by the [conllu\_to\_text.pl](https://github.com/UniversalDependencies/tools/blob/master/conllu_to_text.pl) script from the tools repository.

Example:

# newdoc id = mf920901-001

# newpar id = mf920901-001-p1

# sent\_id = mf920901-001-p1s1A

# text = Slovenská ústava: pro i proti

# text\_en = Slovak constitution: pros and cons

1 Slovenská slovenský ADJ AAFS1----1A---- Case=Nom|Degree=Pos|Gender=Fem|Number=Sing|Polarity=Pos 2 amod \_ \_

2 ústava ústava NOUN NNFS1-----A---- Case=Nom|Gender=Fem|Number=Sing|Polarity=Pos 0 root \_ SpaceAfter=No

3 : : PUNCT Z:------------- \_ 2 punct \_ \_

4 pro pro ADP RR--4---------- Case=Acc 2 appos \_ LId=pro-1

5 i i CCONJ J^------------- \_ 6 cc \_ LId=i-1

6 proti proti ADP RR--3---------- Case=Dat 4 conj \_ LId=proti-1