Studying MWE annotations in treebanks with INESS Search

Course notes for the tutorial at the 2nd PARSEME Training School, La Rochelle 2016

Lecturers: Victoria Rosén and Koenraad De Smedt, University of Bergen, Norway

Technical support: Paul Meurer, Uni Research Computing, Bergen, Norway

Acknowledgments

These course notes have been made possible through the support of the University of Bergen, the PARSEME COST Action, and the INESS and CLARINO projects.

Introduction

This course is targeted at linguists with or without a computational background. The general objective is to enable researchers to search for annotated expressions in various types of treebanks, including all Universal Dependency treebanks, and to create frequency tables for query results. The course method is mainly based on providing a range of typical examples of treebank search.

There are good reasons for using treebanks when studying multiword expressions (MWEs). Treebanks are corpora which are annotated at the level of syntactic structure. They sometimes contain examples of authentic expressions which may be missing in dictionaries.
Lexical units are presented and analyzed in their syntactic contexts. MWEs which are flexible (e.g. phrasal verb constructions) will often be found in different forms in treebanks (e.g. *look up something*, *looked something up*, etc.).

Several treebanks annotate various types of multiword expressions as such. In addition, most treebanks contain useful information for retrieving *potential* multiword expressions.

There are several tools and online services through which treebanks can be queried. In these course notes we only use one such service: INESS, which currently (i.e. 2016) provides treebanks for 51 languages from different sources. MWEs are annotated in varying degrees in the various treebanks.

Treebanks in INESS are searchable with basically the same query language, although specific queries may of course vary with the annotation that is adopted for the different treebanks. This search system is called INESS Search.

Treebank search is a bit more complicated than simple searches for keywords in corpora or on the web. This is because there are possible combinations of several relations, basically dominance and linear precedence. Dominance is the relation between a head and its dependents (in dependency treebanks) or between a phrase and its constituents (in constituency treebanks). Linear precedence is the order in which words appear in the sentence. The INESS Search system allows one to search for combinations of dominance and precedence relations, as well as various restrictions on the nodes.

About INESS and its documentation

INESS stands for "**IN**frastructure for the **E**xploration of **S**yntax and **S**emantics". It has been built by the University of Bergen and Uni Research Computing, with support from the Research Council of Norway. It is part of the CLARINO Bergen Centre and thereby it is part of the larger CLARIN initiative. More information about the project can be found on its website, where a list of related publications can be consulted. The website also contains a substantial amount of documentation, of which we mention the following in particular:

- Grammar documentation: only for some LFG grammars used in LFG parsebanks
- Query language (INESS Search): general documentation, a basic walkthrough, and a page with example searches in the Norwegian treebank (the latter in Norwegian only).
- Web interface documentation: the most relevant parts are the documentation of XLE-Web (the online parsing system), downloading (export) and keyboard shortcuts.
- FAQ
- User Forum

INESS is a recognized CLARIN Knowledge Centre (K-centre) together with the CLARIN LINDAT centre in Prague.

Walkthrough

We suggest that beginners follow the INESS Search Walkthrough, which provides a gentle step by step tutorial to get started.

After the walkthrough you should be able to log in, select one or more treebanks, and perform some basic queries. Note that it is not necessary to log in for accessing many of the treebanks. Test your knowledge by trying the following simple queries. Choose the English UD treebank (eng-ud-1.3-dep) and try the following queries on the Sentence Overview page. By the way, you can only search in treebanks which are indexed for search.

- **[word="work"]**
  This expression looks for all sentences containing nodes having a *word* attribute with the value *work*.
- **"work"**
  This expression is a simplified form of the previous one, because *word* is the default attribute.
- **#x >dobj "work"**
  Search for edges (dependency relations) from any node to a dependent word *work*. The variable *x* stands for any node. Variables are marked by their prefix #.
- **>dobj "work"**
  Simplified form of the previous expression. If a node specification is omitted, a variable is implicitly created. Note however the difference in the way the results are displayed on the Sentence Overview page and on the Query page. If you want a frequency table when you search on the Query page, you must include the variable.
- **>dobj [lemma="work"]**
  Search for an edge labeled *dobj* from any node to a node with the lemma *work*. In other words, search for *work* as a direct object.

- **>nsubj [lemma="work"]**
  Nodes with the lemma *work* as a subject.

You may wonder how to know which edge labels (such as *dobj*) are defined for a treebank, and which node attributes (such as *lemma*) there are and their possible values. Find out by clicking on *Treebank Details*.
A more complex query with combinations of restrictions on nodes is the following:

- **#x:[_pos="VERB" & lemma] >dobj #_y:[pos="NOUN" & lemma="work"]**
  Try this on the Sentence Overview page and on the Query page.
  It matches all sentences with a node for which the *pos* (part of speech) attribute has the value *verb* and which has a direct object where the *pos* is *noun* and the *lemma* is *work*. On the Query page, only the lemmas of the verbs are listed. Variables or attributes containing an underscore are not listed in the table.

In the Spanish UD v1.3 treebank, find all nouns which have no dependent node, in other words, bare nouns, without determiners:

- **#x:[_pos="NOUN" & lemma] & !(#x > #y)**
  Find all nouns *x* (and list their lemmas) for which there is no edge to a node *y*.

The remainder of this tutorial will assume this competence and extend it through examples involving MWEs.

PARSEME wiki table of MWE annotation in treebanks

In cooperation with the INESS project, the PARSEME Action has conducted a survey of how MWEs are annotated in various treebanks (many of which, but not all, are available in INESS). The result (which is still being extended) is documented as a Wiki table on the INESS site, and is useful background information for the present tutorial. When you want to search for a certain type of MWE in a treebank, you may be able to find out how the MWE is annotated by consulting the wiki table.

For a description of this effort, see the following publication:

> Victoria Rosén, Gyri Smørdal Losnegaard, Koenraad De Smedt, Eduard Bejček, Agata Savary, Adam Przepiórkowski, Petya Osenova and Verginica Barbu Mititelu. A survey of multiword expressions in treebanks. In Markus Dickinson, Erhard Hinrichs, Agnieszka Patejuk & Adam Przepiórkowski, editors, *Proceedings of the Fourteenth Workshop on Treebanks and Linguistic Theories (TLT14),* pages 179–193, Warsaw, Poland. Institute of Computer Science, Polish Academy of Sciences.

Annotation types in treebanks in INESS

This section focuses on searching for structures which may be relevant to MWE research.

Dependency treebanks

Dependency structures have labeled directed edges between words and do not directly represent phrasal or clausal units. See the UD row in the PARSEME wiki table for information on how MWEs are annotated in UD treebanks.
Choose the French UD v1.3 treebank. In UD treebanks, fixed expressions are annotated with edges (i.e. dependencies) having the value *mwe*. Check the possible values of the *edge* attribute by clicking on *Treebank Details*.

Go to the Sentence Overview page. Search with the following query:

- **>mwe**
  This expression searches for all *mwe* relations. Click on the first matching sentence and see how the nodes connected by the *mwe* relation are highlighted in red. Click on *Linear view* for a different visualization.

Suppose one wants to search for dependency relations to or from specific words.

- **>mwe "lieu"**
  Dependency relation *mwe* from any node to the word *lieu*.
- **"à" >mwe**
  Dependency relation *mwe* from the word *à* to any node.

Now choose the Query page from the left menu. Try the following:

- **#x >mwe #y**
  The result is displayed as a table showing frequencies for the values of both variables. Note that you need to explicitly name the variables to see them in the table. You can click on a line in the frequency table to see the corresponding sentences.
- **>mwe #y**
  Equivalent to the following:
- **#x_ >mwe #y**
  If a variable is omitted or its name includes an underscore, it will not be included in the frequency table.

Constituency treebanks

See the row for the TiGer and UZH Alpine treebanks in the PARSEME Wiki table. These treebanks provide constituent analyses with labeled edges.
Select the German TiGer constituency treebank (deu-tiger-con). Search for light verb constructions with the edge label *CVC* (which stands for collocational verb constructions) on the Sentence Overview page. *CVC* relations are normally used between an S or a VP and a PP in such a construction.

- **>CVC**
  This searches for light verb constructions. The nodes at either end of the *CVC* edge (normally a *VP* or an S and its dependent *PP*) are highlighted in every match.
- **#x >CVC #y & #x >HD #z**
  This searches for nodes having both a *CVC* relation and an *HD* relation. This means that the *S* or *VP* node as well as the node containing the light verb and the *CVC*-marked node are highlighted.

Search for phrasal verb constructions involving separable verb prefixes with the edge label *SVP*.

- **>SVP "um"**
  When this search is performed on the Sentence Overview page, it gives a list of all sentences containing phrases which contain *um* as a separable verb prefix.

What if we want to make a frequency list of all verbs which contain *um* as a separable verb prefix? Such verbs occur as *HD* (head) of phrases where *um* occurs as *SVP*. Try the following on the Query page.

- **#x_ >SVP "um" & #x_ >HD #y**
  This finds sentences which contain phrases with an *SVP* edge to *um* and which also contain an *HD* edge to a node (variable y). Recall that *word* is the default attribute, so the values for the word forms of the terminal nodes are given.
- **#x_ >SVP "um" & #x_ >HD #y:[lemma]**
  Use this variant if you want the value of *lemma* instead of *word*.

LFG treebanks

LFG distinguishes between different levels of structure. The c-structure provides a constituent analysis with unlabeled edges, while the f-structure is an attribute-value matrix representing relations and features. Look at the PARSEME wiki table row for NorGramBank to see what the analyses look like.

Particle verb constructions can be found by looking for the category *PRT* (particle). As an example, select one of the NorGramBank treebanks, for instance *nob-lbk-tv,* and write the following search expression on the Sentence Overview page:

- **PRT**
  This retrieves all sentences containing a node with the category *PRT*, in other words, nodes containing particles. The *PRT* category is only used with particle verbs in NorGramBank, so these sentences all involve particle verbs.
- **PRT > "ut"**
  This retrieves all sentences in which a node *PRT* has an immediate dominance relation (unlabeled edge) to a node for which the word attribute has the value *ut*. In other words, all occurrences of the word *ut* as a particle. (The word *ut* also occurs with the categories adverb and preposition in Norwegian.)

Fixed expressions are represented in NorGramBank as words with spaces, such as *i dag* ‘today’. In order to find a fixed expression you can therefore simply search for words containing at least one space. This requires a regular expression.

- **".* .*"**
  Finds words starting with zero or more (*, the Kleene star) arbitrary characters (.), followed by a space, followed by zero or more arbitrary characters; in other words, words containing a space.

In the f-structure, the meanings of particle verb constructions or idiomatic expressions are represented by special predicates. These have names composed of the MWE parts, e.g. *cut#down* in the English Pargram treebank (eng-pargram). Select this treebank.

- **'.*#.*'**
  Finds predicates containing #. Note that since you are now searching in the f-structure, and not the c-structure as in the previous example, you must use single quotes around the regular expression.

HPSG treebanks

HPSG treebanks have fairly complex trees with much information on the nodes, including items which may represent noncompositional readings. Look at the information for English DeepBank in the PARSEME Wiki table, in particular for phrasal verb constructions. Intransitive verbs with particles can be found at nodes with *type="v_p_le"*. Select eng-deepbank and write this search expression on the Sentence Overview page.

- **[type="v_p_le"]**
  This will find sentences with nodes representing intransitive particle verbs. As an example, sentence #4 contains a highlighted node with the verb *show*. Notice that this

node contains the reading *show_up_v1*, which is the noncompositional reading of *show* when combined with the particle *up*. This resembles the treatment in NorGramBank.

- **[type="v_p.*_le"]**
  This finds all particle verbs, not only intransitive ones. Again we see the use of the Kleene star in a regular expression. Clicking on sentence #75, we see that the V node dominating the verb *hauled* has the type v_p-np_le. The *.** in the search expression allows zero or more instances of any character, so that we can find both transitive and intransitive particle verbs with this expression.´

Mixed type treebanks

The Lassy Klein (Lassy Small) treebank for Dutch is an example of a mixed type treebank. It has nodes for clauses and phrases with labeled edges, like constituency treebanks, but the graphs do not necessarily maintain linear precedence, like dependency treebanks. Note that access to Lassy Klein requires permission from the rights owners.
Select Lassy Klein (nld-lassy-con) and try the following.

- **mwu**
  This finds sentences which contain a node with the category *mwu* (multiword unit).
- **mwu >mwp [pos='name']**
  This finds sentences which contain a node with the category *mwu* with an edge labeled *mwp* to a node for which the *pos* is *name*; in other words, multiword names.

Sequences and chains of dependency relations

Consider Italian constructions of the kind in the following example, in which the idiomatic expression *dare per scontata* means ‘take for granted’:

Non si può più dare per scontata questa percezione.
‘You can no longer take that perception for granted.’

We know that there are other similar combinations with *dare per* together with a participle or adjective, and we want to find such constructions in a treebank. The Italian UD v1.3 treebank does not explicitly annotate idiomatic constructions. One can either search for these words occurring in sequence, or search for them in a particular syntactic relation.
Select the Italian UD v1.3 treebank and try the following in Sentence Overview.

- **[lemma="dare"] . "per"**
  This looks for the word *per* linearly preceded by a node with the lemma *dare*. However, the results contain occurrences of these words not belonging to the idiomatic expression.
- **"per" . "scontat(o|a)"**
  To check how such constructions are represented, we can try to search for the strings *per scontato* or *per scontata*, literally. Notice the use of alternatives in the regular

expression.
This results in three examples of the idiomatic construction. We find two kinds of annotations, which we try out with the following queries.

- **[lemma="dare"] >nmod #x >case "per"**
  This query illustrates chains of immediate dominance.
  It results in 9 matching sentences, some of which have the idiomatic construction while others show the same annotation for an entirely different construction.
- **[lemma="dare"] >advcl #x >mark "per"**
  This results in 5 matching sentences; again, some of these have the idiomatic construction while others show the same annotation for an entirely different construction.
- **[lemma="dare"] > #x > "per"**
  By omitting the labels on the edges, we get 14 results, i.e. the results of both previous queries.
- **[lemma="dare"] > #x:[lemma & pos & morph] > "per"**
  On the Query page, we get an overview of lemmas involved in this construction, with their attributes *pos* (parts of speech) and *morph* (morphological features). Disregarding irrelevant examples, we see that some of the participles in the construction are annotated as verbs, some as nouns, and some as adjectives.

Searching in several treebanks simultaneously

In UD treebanks, multiword names are annotated with *name* edges. Suppose one wants to search for all multiword names in several UD treebanks at once. Choose several treebanks, e.g. the English, French and German UD v1.3 treebanks. Choose the Query page and use the following query:

- **#x >name #y**
  Note that even if you are viewing one treebank at a time, you can search in several treebanks at once. To the right of the Query field, you see a list of treebanks being searched.
- **#x >name #y :: lang**
  By adding the metavariable *lang*, we see the distribution per language. Notice the edge direction *New* -> *York* for French while German has *York* -> *New*. More about this later.

Some more advanced queries

Dominance relations (edges) can easily be combined with linear precedence restrictions. Suppose we want to search for discontinuous particle verb constructions in the English UD v1.3 treebank, e.g. *Check these out.*

- **#v >compound:prt #p**
  This searches for dominance only.

- **#v >compound:prt #p & !(#v . #p)**
  The dominance relation is combined with a negated immediate linear precedence: the head node may not immediately precede the dependent one. In other words, at least one word must intervene between *check* and *out*.

More frequency tables

Select some UD treebanks and execute these searches on the Query page.

- **#x_ >mwe #y_ :: lang**
  Edges labeled *mwe*, totals per language
- **#x_ >mwe #y_ &!(#x_ >mwe #z & #z != #y_) :: lang**
  Two-word fixed expressions: nodes which have only an *mwe* edge to one other node
- **#x_ >mwe #y_ & #x_ . #y_ &!(#x_ >mwe #z & #z != #y_) :: lang**
  Head-initial *mwe* relations involving only two words: the dominant node must precede the dependent one
- **#x_ >mwe #y_ & #y_ . #x_ &!(#x_ >mwe #z & #z != #y_) :: lang**
  Head-final *mwe* relations involving two words: the dependent word must precede the dominant one
- **#x >mwe #y & #y . #x &!(#x >mwe #z & #z != #y) :: lang**
  Head-final *mwe* relations involving two words, showing words
- **#x >mwe #y & !(#y . #x) & !(#x . #y) &!(#x >mwe #z & #z != #y) :: lang**
  Edges labeled *mwe* between non-adjacent words
- **#x >mwe #y & #x >mwe #z & #y .* #z**
  Fixed expressions with at least three elements (assuming *x* is the head of both)
  With *#y .* #z* we require that *#y* comes before *#z*, otherwise the expression is symmetric and we get double matches.
- **#x >mwe #y & #y >mwe #z**
  This finds incorrectly annotated expressions in the German UD v1.3 treebank where a word *y* is both the dependent of an *mwe* edge and the head of another *mwe* edge.

Note: it is not possible in INESS Search to return a list of all MWEs containing an indefinite number of words, because a variable can only be bound to a single node, not a sequence of nodes. Another search system, PML Tree-Query at LINDAT, has mechanisms for repeating operations on a number of nodes.
Note: It is however possible in INESS Search to match nodes with a specific number of children with the predicate _arity(#x,n) or to specify minimum and maximum distances in the dominance operator >{m,n} and the linear precedence operator .{m,n}.

Some UD treebanks have incorrect or even inconsistent annotation of which node is the head of *mwe* and *name* edges. Select all treebanks in the UD 1.1 collection and perform a small check on the Query page.

- **#x:[word="New|York"] >name #y:[word="New|York"] :: lang**
  This matches all combinations of *New* and *York*, head initial or head final, and lists frequencies for each combination per language.

Suppose we want to search for multiword prepositions such as *by way of, with respect to,* etc. In NorGramBank these are normally annotated as words with spaces, e.g. *i nærheten av* (near). Select one of the NorGramBank treebanks.

- **P > #x:".* .*"**
  This searches for a node with the category *P* (preposition) immediately dominating a word that contains a space.
- **PP -> #x_:P #y_:NP & #y_ -> #z_:N #u_:PP & #u_ > #v_:P & #x_ > #a:[value] & #z_ > #b:[value] & #v_ > #c:[value]**
  This matches preposition-noun-preposition sequences that are coded as regular *PP*s embedded in *NP*s embedded in *PP*s, and are not likely to be multiword prepositions.

In UD treebanks, multiword prepositions are not explicitly annotated. We could search for patterns such as *preposition-noun-preposition* that are MWE candidates. Select the English UD v1.3 treebank and perform these searches on the Query page:

- **[pos="ADP"] . [pos="NOUN"] . [pos="ADP"]**
  This searches for linear precedence only, and will only give the total. The result includes many irrelevant matches.
- **#x:[_pos="ADP" & word] . #y:[_pos="NOUN" & word] . #z:[_pos="ADP" & word]**
  This will show the frequencies of word combinations.
- **#x:[_pos="ADP" & word] . #y:[_pos="NOUN" & word] . #z:[_pos="ADP" & word] & #y >case #x & #y > #_c >case #z**
  Includes a restriction with case edges, to select more suitable candidates.
- **#x:[_pos="ADP" & word] . #y:[_pos="NOUN" & word] . #z:[_pos="ADP" & word] & !(#y >case #x & #y > #_c >case #z)**
  Negates the previous restriction with case edges, i.e. these are probably unsuitable candidates. There are two sentences where *in case of* is annotated with a *mwe* edge from *in* to *case*. Of course *in case* can be a multiword expression on its own, for example in *in case they don’t come*, but *in case of* should probably be annotated as a separate MWE.

End.