Mapping FinnTreeBank to Universal Dependencies

Jussi Piitulainen, Hanna Nurmi
Department of Modern Languages, FIN-CLARIN
University of Helsinki

March 2016

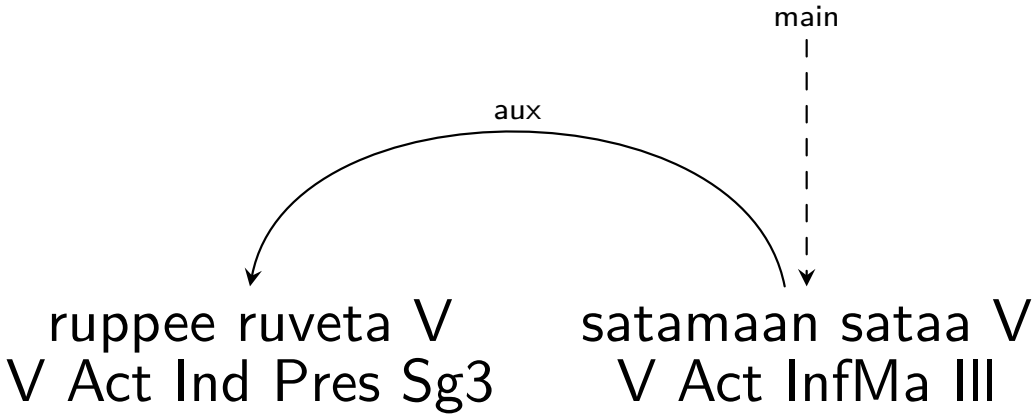
Table of Contents

Native FinnTreeBank model

- FinnTreeBank 1 (FTB1) as a Grammar Definition Corpus
- VISK example sentences, hand-annotated
- Annotation model by Atro Voutilainen and group
- In connection with FIN-CLARIN
- Licensed as a Free Cultural Work under CC BY 4.0, and as Free Software under LGPLv3+, at your option.

FinnTreeBank annotations

- Each sentence is segmented into tokens.
- Each token has a lemma and a coarse class (known as POS)
- and a finer multi-component label also as features
- and a labeled dependency to its head.



- POS=VERB,VOICE=ACT,MOOD=INDV,TENSE=PRE...
- POS=VERB,VOICE=ACT,INF=MA,CASE=ILL

FinnTreeBank principles

- Focus on “content words” as heads
- Deep structures favoured over flat
- Verb chain as unit (starting with subject, then auxiliary verbs, modal nouns and adjectives, ending with main verb)
- flat coordination by chaining (UD uses bundling)
- chaining of attributes now considered a mistake
- parsability? underspecificity? informativeness?

Universal Dependencies model

`http://universaldependencies.org`

Cross-lingual project

A growing collection of treebanks with a common annotation model being developed and revised for diverse languages on GitHub by Joakim Nivre, a core group, and a cast of interested parties.

- More languages join, revision released every six months.

Components of a description

Segmentation into tokens with lemmas, fixed set of "parts of speech" labels (coarse word classes, Google), more flexible morpho-syntactic descriptions (key-value pairs, Prague), labeled dependency relations between tokens (Stanford).

Universal Dependencies goal

`http://universaldependencies.org`

Use the same model for different languages

A major goal is to find a unified way to describe syntactic structures of different languages.

- Use the same annotation to describe the "same thing".
- Use different annotations to describe "different things".
- Document language-specific choices and extensions.

Comparability?

This might make it more meaningful to compare the state of art across languages.

Technology

Universal Dependencies

The model is developed by a machine learning community (CoNLL).

FinnTreeBank

- Omorfi has been adapted
- tagger?
- (shallow) parser being developed?

Turku Dependency Treebank

Finnish UD model

The UD project requires a language-specific annotation manual, Turku Dependency Treebank (TDT, Filip Ginter, Sampo Pyysalo, group) provides a "main" UD treebank and manual for Finnish.

Three targets for FinnTreeBank

When to follow what?

- Generic UD model (general principles)
- Finnish UD model (from Turku)
- Native FinnTreeBank model

When may we follow what when we map?

Maybe follow what seems best from our point of view?

Prefer generic UD to Finnish UD when generic UD agrees better with native FinnTreeBank model (or VISK) or seems otherwise better to us.

Example: both UD and FTB (with VISK) have a particle class

So particles may be considered good to have even against a Finnish UD (Turku) decision.

- The FTB particle class has subclasses; some are mapped to more specific UD classes.

Example: may follow generic UD with determiners

Because UD determiner class agrees with determiner relation

So our mapping may favour generic UD analysis over current Finnish UD for certain uses of pronouns as determiners.

FTB	tämä PRON attr
Fi-UD	tämä PRON det
UD	tämä DET det

FTB	minun PRON attr
Fi-UD	minun PRON nmod:poss
UD	minun DET det

Relational examples of when to follow what

Our mapping agrees with UD against Fi-UD

expl (expletive or pleonastic nominal, in FTB also adverb)

UD agrees with Fi-UD against our mapping

auxpass, parataxis (sentences joined with :;- or sentences with reporting clause)

Our mapping agrees with Fi-UD against UD

Use xcomp:ds, do not use iobj (indirect object), do not use dislocated, do not use list, common restrictions on subject complements (cannot be adpositional phrase).

Principle on in FTB

The head of a phrase is always a content word:
kuppi **kahvia**, joku **pojista**.

Mapping and swapping

Overall architecture

First a "mapping" script relabels classes, features, relations; then a "swapping" script adjusts tree structures.

```
./finntreemap [options] < ftb1-raw.tsv |
./finntreeswap [options] > ftb1-ud.conllu
```

Mapping script

Walks each token, decides on new labels of all types based mainly on old features of token and its parent token.

Swapping script

Recognizes specific local configurations, typically makes old dependent new head and adjusts many links

Mapping

Even mere relabeling of things is surprisingly messy.

for each token:

if a condition holds: set result class, features, relation

else if a condition holds: set result class, features, relation

...

else: set catchall result class, relation.

Have *access* to whole source tree, *easy access* to current token's annotation (including relation to head) and its head token.

Sensitive to order

More specific conditions first.

Catch frequent mappings first and somewhat well

Accept some leakage.

Swapping

Structural changes

are more tedious to program than (mere) relabelings.

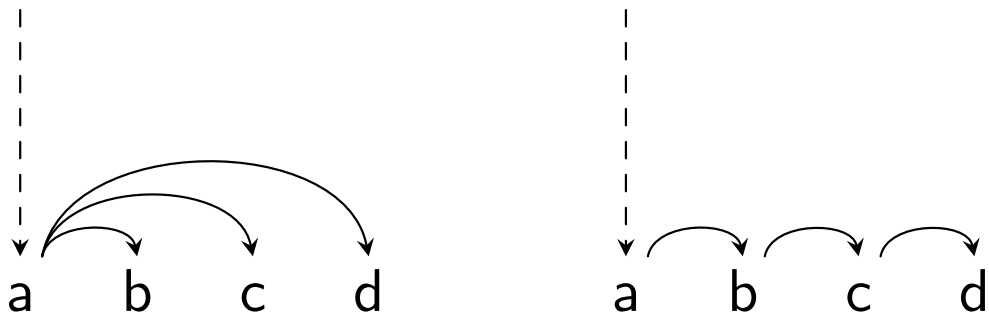
For example, in copular clauses

swap the verb and its complement (scomp becomes head, copula becomes cop), and raise all dependents of the copula to the head. Maybe.

- other verb chains
- coordination
- punctuation marks should link higher (raw FTB does not link them anywhere)

Structural tendencies

- UD is bundling (flatter)
- FTB is chaining (deeper) (verb chains, coordination, idioms)



Examples follow

Much is easy to map

Often straight relabeling (of classes, feature keys and values, relations) works.

FinnTreeBank alternative annotation format helped

Key-value pairs originally developed to match Omorfi, now sometimes easier to match with UD features than the native label sequences.

Much is also not

So the examples that follow are examples of problems.

Tokenization

Multi-word names

FinnTreeBank used to have multi-word names as single tokens. These should remain mappable when split into multiple tokens, but FinnTreeBank does not have a special name relation! And attr in source would be ambiguous, leading to complex or fragile conditions?

What on earth about Trunc

FinnTreeBank has truncated parts of compound words as a word class.

Compounding with the negative verb

A number of rather special compounds in Finnish

Certain adverbs and things compound with the negative verb.
Similar to English “wouldn’t” for “would not”.

In FinnTreeBank tokenized as two

Ellen = Jos en; mikset = miksi et; etc. These inflect, too.

```
3  ell   jos   Pcle
4  ei    ei    V
```

In Finnish-UD as one

```
3  ellei   jos#ei   V
```

Splitting labels based on conditions

Relabeling is sometimes messy.

FinnTreeBank classes need splitting

- N goes to NOUN or NPROP (when proper)
- Pron, Pcle, Adp, Adv go to wherever by whatever criteria (ok, Adp is easy to map; some Pcle go to CONJ)
- What goes to DET?

As a last resort

Map to X (catchall class), but rather not.

Mapping FinnTreeBank relations

FinnTreeBank relations need splitting

- subj goes to nsubj (when N), clsubj (when V)
- phrm goes to cc, mark, expl (by whatever criteria)
- advl goes to ccomp, xcomp, advcl, advmod, nmod, discourse.

As a last resort

When all else fails, map to dep (catchall relation).

Features

Often simple renaming

FTB InfMa has lative case marked (because translative is also possible). This is language specific but can stay.

But not always

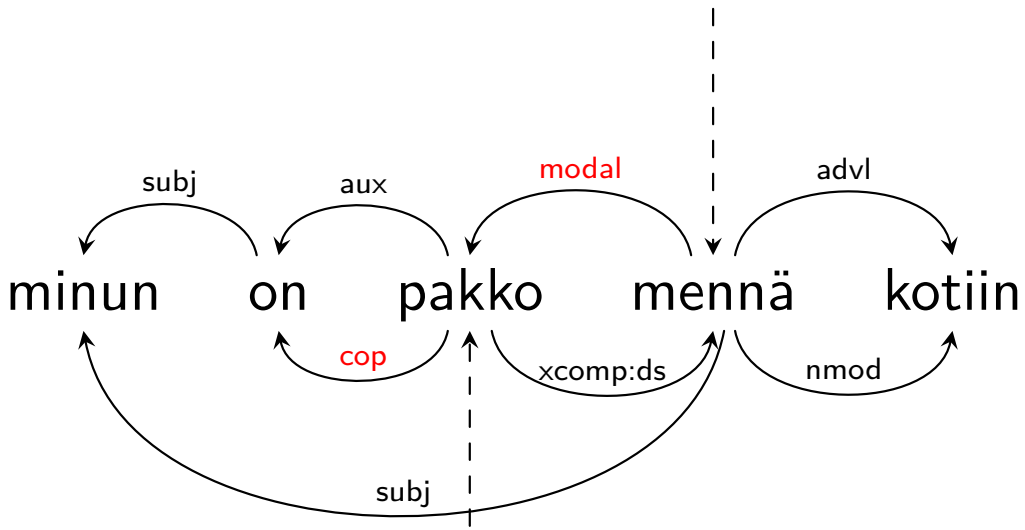
FTB does not label positive degree explicitly, these need to appear.

Pronoun types

UD has “indefinite” and “total” pronoun types Ind and Tot but no “quantifiers”, Fi-UD has only Ind (includes Tot), FTB (with VISK) has Qnt (includes Tot), we must map Subcat=Qnt to PronType=Ind.

modal

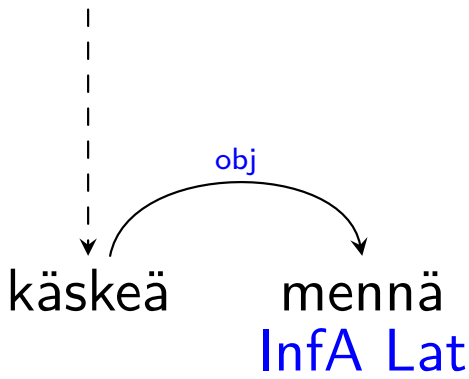
FinnTreeBank verb chains can contain nominal components (like pakko), labeled as modal, with a possible genitive subject.

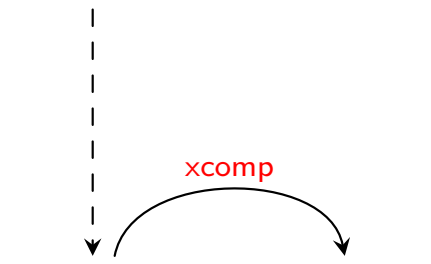


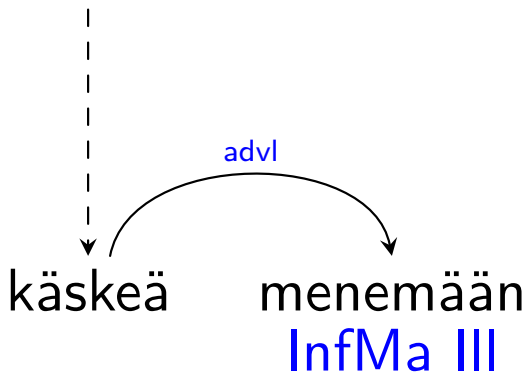
These are now mapped manually (in FTB1) to copular structures – a structural change – according to the Turku model. (Nominal part was mismapped to aux before.)

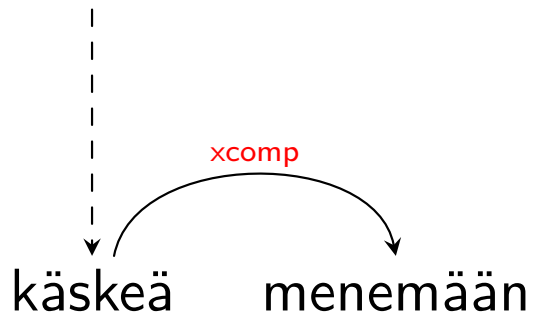
Mapping to xcomp is trouble

In UD, the subject of an xcomp is *determined* by its head, else the verbal complement is ccomp. FTB *does not make this distinction* but makes other, more formal distinctions. Obligatoriness?









The extent of the xcomp trouble

Many occurrences, currently all mismapped

FinnTreeBank 1 has at least 1300 verbs as obj or advl that *should* be mapped to xcomp but *are now* either ccomp or advcl. Some others should be xcomp:ds in Finnish UD.

Identified by hand (well, eye, maybe)

The 1300 xcomp instances (so far) in FTB1 found were identified by a human expert. We don't know how to automate this! Yet?

Why doesn't FinnTreeBank model mark this?

Because not done *yet*? Because obligatoriness and control not otherwise important in FTB? (FTB decides between obj and advl by infinitive type and case.) Because it would not be decidable?

Treatment of copulas

Universal Dependencies lowers copulas

Copulas mostly as dependents (because overt copulas may not even be there).

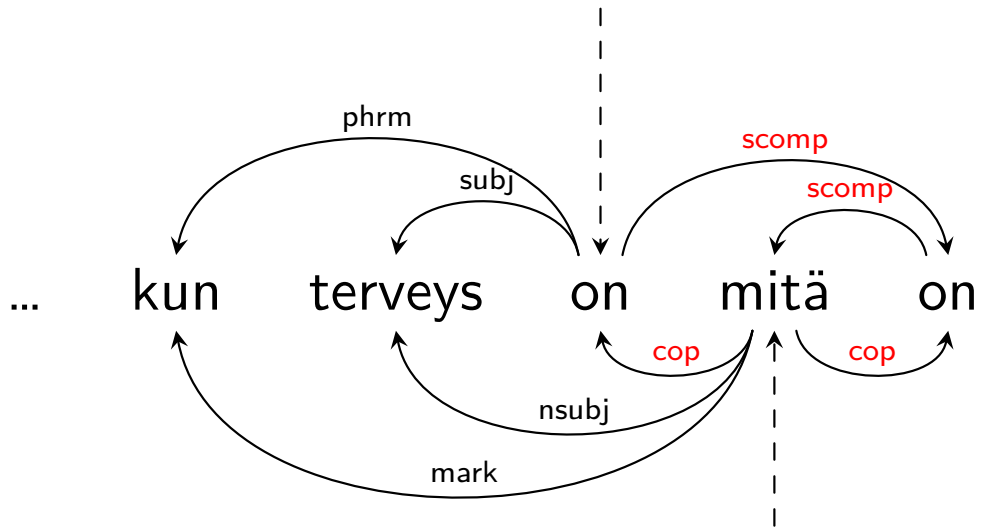
Nesting turns out to be messy

Nested copular sentences become messy when copula is not head, so UD backtracked on problem cases: now the copula may be head after all.

- "Inconsistent and ugly"
- "Turku has an alternative"

UD copulas

Problem when complement is a clause becomes acute when it is a copular clause.



(Hanna interprets generic UD docs possibly this way, Turku has a different analysis with a relative clause.)

Remaining family of sentences types

Possessive, existential, experience sentences

We haven't mapped certain sentence types properly.

- Minulla(subj) on kissa(obj).
- Pöydällä(nmod) on kissa(subj).
- Minulla/minun(subj) on kylmä(scomp).

(Rule: subject in adessive, verb olla, has obj, map as above?)

Surprised by the machine learners

Sentences need partitioned for machine learning

The community wants each treebank in three parts, in a standard methodology for training statistical models:

- training set,
- development set,
- test set.

They want the partitioning stable across releases

Sentence must *stay in the same set* when UD FinnTreeBank is revised.

So each sentence more or less needs a persistent identifier

A good idea anyway. These need added.

What now, where next?

What is worth doing (Chesterton)

is worth doing badly.

Dealing with failure is easy. (Perlis)

Work hard to improve.

Dealing with success is also easy. (Perlis)

You have solved the wrong problem. Work hard to improve.

Summary: yeah, worth it.

Both successes and failures in the mapping to a cross-lingual model may reveal something interesting about languages, and about the models.

- Also, successes produce useful tools and data sets.