Understanding syntactic change: Constructing the infrastructure

Beatrice Santorini, University of Pennsylvania

Syntactic theorizing vs. syntactic annotation

- As syntacticians, we get to choose problems of particular interest to us and to ignore other phenomena, even common ones.
  - Coordination
  - Amalgams (That's all they do is fight)
  - Expressions of time
    - They disappeared a year ago.
    - Ils sont disparus il y a un an.
    - Una vocе poco fa qui nel cor me resuonÀ!
  - Disfluencies
- As annotators, we don't get that privilege.

Correctness vs. practicality

- Given our (partial) ignorance, annotations can't always be correct.
- But they don't need to be.
- They need to be practical.

Practical for whom?

- An annotated corpus needs to be practical
  - for the annotator (during corpus construction, revision, and maintenance)
  - for the user (during searches)
- These two aspects are potentially at odds with each other.

Practicality for the annotator

- Interface issues (text vs. graphical display, ...)
- Annotation guidelines need to be easy to implement in terms of speed and consistency.
- Set of categories (size of tagset, names and abbreviations of categories, ...)
  - Brown Corpus (1M words): ca. 90 tags
  - Penn Treebank (5M words, show feasibility for 100M): ca. 50 tags
- Avoid difficult decisions
  - Ignore distinction between verbal and adjectival participles
  - Omit VP
- Use defaults
  - Attachment ambiguity
    - High attachment in PPCHE, following Penn Treebank
  - Position of traces
    - Clause-initial (impossible but useful)
  - Basic phrase structure
    - Default VO + leftward movement for Old French
  - Old vs. new grammar
    - Assume *do* support only after 1500
    - Prefer analysis as object control over ECM

Practicality for the user

- Distinguish same vs. different (say, object control vs. ECM)
- Intuitive annotations might be diachronically wrong.
  - Historically correct annotation:

```
(IP-ABS (NP-SBJ (D a) (ADJ long) (N time))
        (VBN ago(ne)))
```

  - Synchronically more intuitive annotation:

```
(ADVP-TMP (NP-MSR (D a) (ADJ long) (N time))
          (ADV ago(ne)))
```

  - Other possibilities:

```
(ADVP-TMP (IP-ABS (NP-SBJ (D a) (ADJ long) (N time))
                  (VBN ago(ne))))


(IP-ABS-TMP (NP-SBJ (D a) (ADJ long) (N time))
            (VBN ago(ne)))
```

Harmony between the two aspects of practicality

- Avoid structure that makes queries longer, more complicated, more error-prone, ... without any substantive effect on search results
- Current guidelines eschew:
  - Intermediate projections
  - Noun phrases as DPs
  - Detailed left periphery

Tension between the two aspects of practicality

- The interests of the annotator and the user can be at odds with each other.
  - Common vs. proper noun
  - Adverb vs. discourse particle
- It makes sense to resolve tensions in favor of the user.
- Since the user tends to want more information, and more information can be added to the corpus over time, that will be the natural direction that the annotated corpus moves in.

Difficult constructions

- Flag and put aside – in the hopes that the correct solution will become apparent

  Je ne sais que faire.
  I NE know QUE do
  'I don't know what to do.'

  Je ne sais que faire la vaisselle.
  I NE know QUE do the dishes
  ?????

  Je ne connais que toi.
  I NE know QUE you
  'I know only you.'

- Give up and use the labels "X" and "XX"

  When they had finished their work, they left.
  They finished their work, and they left.

  When they had finished their work, (X and) they left.

Normalizing word tokenization

- The same syntactic structures can be associated with more than one orthographic form or convention.
  - Particles ("separable prefixes") in German, etc.
  - Clitic negation and contractions in English, etc.
  - Portmanteaus of preposition and determiner in Romance
  - Unconventional spellings by unschooled writers
- Normalization is in order – that is, word tokenization in accordance with syntactic structure
- A striking example from Marguerite de Valois

  un for beau prÃ© ou il lia des arbres
  'a very beautiful meadow where he linked trees'

  ?????

  un fort beau prÃ© ou il y a des arbres
  'a very beautiful meadow where there are trees'

  un for beau prÃ© ou il=l@r @r@ @a d@r @es arbres

- By the way:

  Qui crois-tu qui est venu?
  who believe you QUI is come
  'Who do think came?'

  Qui crois-tu qu' il) est venu?
  who believe you that he (resumptive) is come

Mise en place

- Professional painters can spend more time prepping the surface than applying the paint.
- And professional cooks have a name for that...
- Before POS tagging
  - Word tokenization
- During POS correction (doesn't need to be consistent)
  - Sentence tokenization
    - The shorter the sentence token, the easier to parse.
  - Add empty categories
    - Pronouns, complementizers, ...

Scaffolding

- Add information to a tagged file that can be deleted in the parsed file
  - Fake punctuation
    - Quotation marks, question marks, ...
  - Add diacritics to POS tags to facilitate parsing
    - Distinguish *that* in ordinary complement clauses vs. relative clauses

Private (working files) vs. public (release) version

- Extending the scaffolding idea, the private version might contain:
  - various types of comments (notes to self, ideas for improvement, ...)
  - partial implementation of certain annotations and distinctions
  - "beta" features
  - "ergonomic" annotation
- Delete or revise the scaffolding information for the public version via a script

An example of "ergonomic" annotation

- Official labels are asking for typos
  - NP-OB1, NP-OB2
- "Ergonomic" labels avoid typos, but are wrong for languages without morphological case
  - NP-ACC, NP-DTV
- As long as mapping between labels is one-to-one, no problem
- Use "ergonomic" labels in the working files and globally replace them in the release version.
- Extending the idea: NP-a, NP-d, ... + global replace

Using CorpusSearch

- CorpusSearch was originally built as a tool for searching completed corpora. But it can also be used to construct and maintain corpora.
- Poor person's parser

  The revision feature of CorpusSearch allows you to build a skeletal parse from a POS-tagged corpus. This is very useful if you are working on a language for which there is no training corpus.

  - A simple example:

```
Input:
(D an) (N example)


query:     ((1)D hasSister (2)N)
        AND (D iPrecedes N)
add_internal_node(1,2)= NP


Output:
(NP (D an) (N example))
```

- Computer-aided correction
  - Early on, annotators corrected the output of an automatic parser sentence by sentence ("narratively").
  - The annotator's attention is constantly switching from one error type and annotation guideline to another.
  - The revision feature in CorpusSearch allows another mode of correction: error type by error type.

    For instance:

    - Search for sentences with two finite verbs
    - Search for sentences where *serve* takes a dative complement rather than accusative
  - The principle is the same as that in Adam Smith's famous description of the pin factory.
  - Errors can be automatically revised (silently or with flagging) or just flagged for completely manual review and revision.
    - I tend to flag for manual review, because there are more things in heaven and earth than are dreamt of in our philosophies.
    - Tony was always in favor of automatic revisions in the interests of speed.
    - A rational approach would be to try to fine-tune the queries to maximize speed and minimize errors.
    - But the rational approach might be more time-consuming than it's worth...
  - Using this approach, annotation involves many passes through the corpus.
  - Ideally, correction boils down to a choice between two alternative annotations.
  - My guess is that this method speeds up corpus construction by something like a factor of 4 over "narrative" correction.
  - Any errors that you notice that are not the current focus of attention can be corrected in passing. Obviously, if an error type occurs often enough, it should give rise to a query of its own.

Correcting entire corpora instead of individual files

- Another application of the pin factory approach is to combine all the files in a corpus and work on the entire corpus.
- Be sure to delimit the files with a comment if you want to split them for the release version.
- Emacs handles the entire corpus of Modern British English or the entire Parsed Corpus of Early English Correspondence.

Post-release errors

- No matter how many sanity checks you perform, there are sure to be errors in the release version.
- Users need to be encouraged to send error reports. Sometimes, they need to be trained to include relevant information, such as a token's ID number.
- Let me pause here to thank Sasha Simonenko for reporting dozens of errors in the historical French corpora.
- If the reported error is not clearly a one-off error, it is sensible to write a CorpusSearch query that will catch the reported error as well as ones like it.

Known issues

Hiring annotators

- Fewer is better (because of interannotator consistency)
- Obviously, a good grasp of syntax is essential
- Comfortable with the notion of notational variation
- Helpful experience includes any skill that depends on repeated effort to attain a result.
  - Knitting
  - Playing a musical instrument
  - Possibly: coding

Ideas for the future

- Partial annotation + cross-project collaboration
- Use CorpusSearch definition files as a poor person's lemmatizer
- .... [your suggestions here]