CorpusExtract: A Tool for Analyzing Syntactically Annotated Corpora

Kenneth Hanson

MSU Language Acquisition Lab

Background

Corpora and Language Change

The English language has changed over time: pronunciation, vocabulary, morphology (word structure) and syntax (sentence structure) are no longer the same as they were centuries ago.

Examples of syntactic change:

- OV to VO (Old English to Middle English)
- [V Adv O] to [Adv V O] (Middle English to Modern English)

Corpora, which are collections of text or transcribed speech, are an important resource for studying language variation and change.

Unannotated corpora are adequate for identifying basic patterns, but in order to draw any quantitative conclusions, such as whether changes proceeded in different contexts at the same rate, linguists must **code** corpora for statistical analysis.

Coding by hand is a painstaking and error-prone process. Fortunately many corpora have already been annotated using semi-automatic tools. For example, the Penn Corpora of Historical English are annotated with part-of-speech tags as well as syntactic trees, making it easier for linguists to study how English syntax has changed.

Automated Coding with CorpusSearch

A tool called CorpusSearch can be used to search texts in the Penn Corpora of Historical English (and similarly formatted corpora) for linguistic structures with certain features. The program also supports coding queries, which mark structures for the presence or absence of such features.

Figure 1 shows one output item from a CorpusSearch query that marks all noun phrases for the presence or absence of quantifiers, determiners, and prepositional phrases. The coding strings (shown in red) have been added to the original syntax tree.

Figure 1: Output from a CorpusSearch coding query

```
/~*
Syttyngge at +te mete, loke sche turne aboute in here
herte +te clennesse of here chastete,(CMAELR3,28.53)
*~/

( (IP-IMP (IP-PPL (VAG Syttyngge)
          (PP (P at)
              (NP (CODING-NP x:d:x) (D +te) (N mete))))
      (, ,)
      (VBI loke)
      (CP-THT (C 0)
          (IP-SUB (NP-SBJ (CODING-NP-SBJ x:x:x) (PRO sche))
              (VBP turne)
              (RP aboute)
              (PP (P in)
                  (NP (CODING-NP x:x:x) (PRO$ here) (N herte)))
              (NP-OB1 (CODING-NP-OB1 x:d:p) (D +te) (N clennesse)
                  (PP (P of) (NP (CODING-NP x:x:x)
                      (PRO$ here) (N chastete))))))
      (. ,)) (ID CMAELR3,28.53))
```

Problem

The output of CorpusSearch is difficult to examine by hand, a step that is typically required when using automated searching of corpora. Additionally, if the user exports the coding strings (as in Figure 2), the original text is lost. This makes it extremely difficult to do additional manual coding.

Figure 2: Output from a CorpusSearch command to print all coding strings

```
x:d:x
x:x:x
x:x:x
x:d:p
x:x:x
...
```

The ideal situation is to have the output of a CorpusSearch coding query formatted as a spreadsheet containing:

1. The text of each coded item
2. The value for each coding variable for each item
3. The sentence containing each item
4. The source line of each item

This is a typical format used during manual coding.

Solution

In order to solve the problems with the output of CorpusSearch, I wrote a small program called CorpusExtract. The program takes in a coded corpus file produced by CorpusSearch, and outputs a tab-separated spreadsheet file that can be viewed in any spreadsheet program (and optionally converted to the native format).

Figure 3 shows a portion of output produced by CorpusExtract from the coded corpus shown in Figure 1.

Figure 3: Portion of a spreadsheet generated by CorpusExtract

…	…	…	…	…	…
here bosum	x	x	x	In mete and drynk, in slep, in spekyngge, eueremoor sche moot drede apeyrryngge of here chastete, an-aunter +tat +gyf sche +gyue moor +tan is due to heore flehs, sche +gyue streng+te to heore aduersarie, and nursche here enemy pryuely in here bosum.	(CMAELR3,27.51)
=+te mete	x	d	x	Syttyngge at +te mete, loke sche turne aboute in here herte +te clennesse of here chastete,	(CMAELR3,28.53)
sche	x	x	x	Syttyngge at +te mete, loke sche turne aboute in here herte +te clennesse of here chastete,	(CMAELR3,28.53)
here herte	x	x	x	Syttyngge at +te mete, loke sche turne aboute in here herte +te clennesse of here chastete,	(CMAELR3,28.53)
here chastete	x	x	x	Syttyngge at +te mete, loke sche turne aboute in here herte +te clennesse of here chastete,	(CMAELR3,28.53)
=+te clennesse of here chastete	x	d	p	Syttyngge at +te mete, loke sche turne aboute in here herte +te clennesse of here chastete,	(CMAELR3,28.53)
=+tat vertu	x	d	x	and inwardliche si+g+gyngge to +te parfeccioun of +tat vertu, let here saade here mete, and o+tur-while haue scorn of here drynke;	(CMAELR3,28.54)
…	…	…	…	…	…

Implementation

CorpusExtract is essentially a **parser**, a program that processes text input that conforms to a language describing the format of the input. It interprets a coded corpus file in the same way a programming language compiler interprets the source code for a computer program.

Parsers and Parser Generators

Many possible languages (a subset of the *context-free* languages) can be parsed by an **LR parser** (where LR refers to a generalized algorithm), with minor variations specific to the grammars of individual languages.

Because of this, it is not necessary to implement a parser from scratch; instead the programmer can use a **parser-generator**, which requires the programmer to:

1. Define the rules describing a valid input file
2. Define any actions taken by each rule in order to generate the output

The parser-generator uses these instructions to produce the source code for a parser, adapting the general algorithm to the language defined by the programmer. The resulting code can then be compiled to create an executable program.

ANTLR

I used the parser-generator ANTLR (v3.5), which produces Java code by default, the same language that CorpusSearch is written in. Compiled Java code runs on the Java Virtual Machine, which is available for all major operating systems, allowing CorpusExtract to be used on almost any computer.

References

CorpusSearch:

http://corpussearch.sourceforge.net

ANTLR 3.5:

http://www.antlr3.org

Funded by the MSU College of Arts and Letters Undergraduate Research Initiative