Methods for Tracking Lexical Classes in Parsed Historical Corpora

Kenneth Hanson
MSU Language Acquisition Lab

Background

Bare Singular NPs in the History of English

The methods presented here were developed for an ongoing study tracing the near complete loss of **bare singular noun phrases** between **Middle English (ME)** and **Modern English (ModE)** [1].

(1) John is a doctor. [ME: OK, ModE: OK]
(2) John is doctor. **[ME: OK, ModE: Impossible]**
(3) John is president. [ME: OK, ModE: OK]

The goal of the study is to ***identify the nature and cause of the change*** by performing a statistical analysis of the changing distribution of bare singulars in different environments. In order to do this, we must obtain a large number of data tokens and **code** the factors that condition bareness:

- ***syntactic position*** (e.g. object of the verb "be") and
- ***lexico-semantic class*** (e.g. unique/non-unique roles) [2]

...as well as ***bareness*** and ***date of source text***.

Searching and Coding Historical Corpora

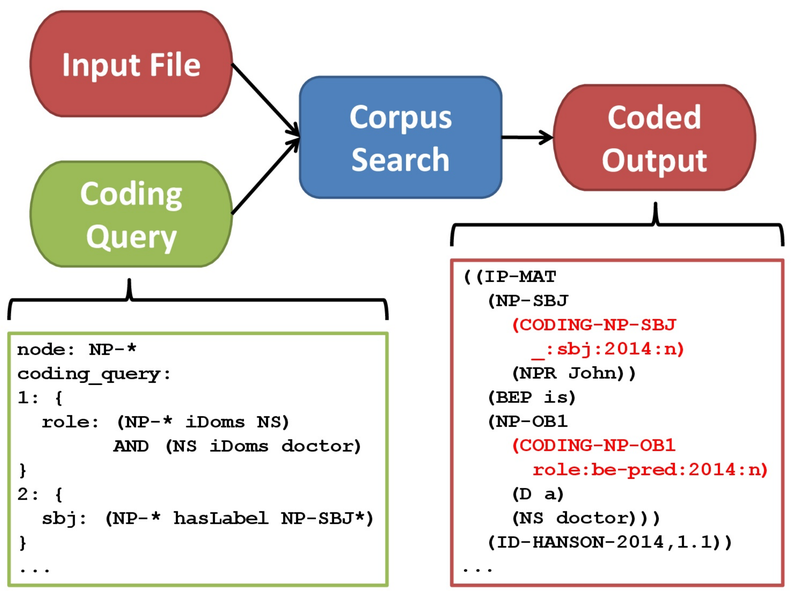
The most effective way to accomplish the above task is to utilize **parsed historical corpora** [3, 4, 5].

- **Historical corpus**: a collection of (digitized) historical texts
- **Parsed corpus**: includes syntactic trees for every sentence

These corpora can be automatically searched and coded using the **CorpusSearch** program [6], as shown in Figure 1. However, the coding scheme for the study is too complex to be executed using the existing functionality of the program, ***requiring the development of additional tools and techniques***.

Problems dealt with include: (i) coding of lexical items and classes, (ii) coding of syntactic position, (iii) dating the source texts, and (iv) managing the complexity of multi-part queries.

Figure 1: Coding a parsed corpus using CorpusSearch



Tracking Lexical Items and Classes

In order to track **lexical classes** (words with similar properties), words need to be chosen to represent those classes. This information needs to be ***stored in a human-readable format*** to ensure correctness and allow easy modification.

Problem 1: Complex CorpusSearch queries are not readable, and modifying information that appears in multiple places is error-prone.

Solution: Store the coding data in well-formatted **databases**, and ***convert to CorpusSearch format*** using a Python script.

Problem 2: English spelling was not standardized before the modern period, and the English historical corpora are not **lemmatized** (i.e. different spellings of the same word are left as is).

Solution: Create two databases, one for the nouns representing the classes, and a second containing spelling variants for every noun. Coding will be done in two steps: head noun first, followed by class.

Coding Syntactic Position

For most noun phrases, the syntactic position is encoded in the node label. We can use this to code for syntactic position.

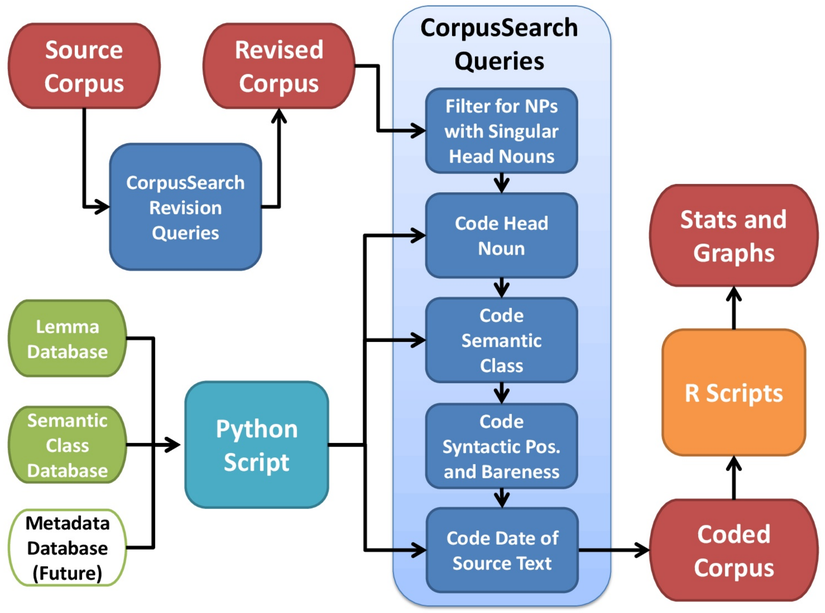
Problem 1: Not all of the position labels needed are included in the annotation scheme.

- Labels provided: NP-SBJ, NP-OB1, NP-OB2, NP-SCP, NP-SPR
- Labels desired: NP-OB1-BE, NP-OB1-HV, NP-POBJ, NP-POBJ-AS

Problem 2: The msot obvious solution is to code for syntactic position based on where the noun phrase occurs in the tree. However, CorpusSearch ***cannot reference the external context*** of a node in search or coding queries.

Solution: Use a **revision query**, which can append new sub-labels based on external context. Then, code syntactic position as usual.

Figure 2: The complete search and coding process



Dating Tokens

In order to track changes over time, we need to record the date of the text each token comes from.

Problem 1: Annotation of dates is dones in different places in different corpora, or may be missing entirely (in the case of ME).

Problem 2: Every possible date for each annotation system requires a separate query file entry.

Solution 1 (Current): Use one giant, complicated coding query.

Solution 2 (Future): Create a **text metadata database**, and convert to CorpusSearch format as with the other databases.

Managing Multi-step Queries

For complex searches, it is often convenient or even necessary to ***run multiple search and coding queries in sequence***. Reasons for this include:

- Limitations of CorpusSearch
- Reduced search time
- Make queries more flexible/understandable

Problem: This makes executing searches more complicated, since intermediate output needs to passed to the next search.

Solution: Use Unix **makefiles** to coordinate the process.

Additional benefits:

- The entire study can be replicated exactly, with no possibility of the researcher mxing up filenames or forgetting commands.
- Searches can be resumed from an intermediate step in the case of errors or changes to the queries.

The complete data flow is shown in Figure 2.

Acknowledgments

This work is part of a larger project with Cristina Schmitt and Alan Munn [1], and is partly funded by the MSU College of Arts and Letters Undergraduate Research Initiative (CAL-URI).

Special thanks to: Beatrice Santorini (CorpusSearch), Aaron Ecay (text dating), and Joel Wallenberg (text metadata file).

References

[1] Kenneth Hanson, Cristina Schmitt, and Alan Munn (2014). The Loss of Bare Singular Arguments and Predicates in the History of English. To be presented at the 16th Diachronic Generative Syntax Conference. Research Institute for Linguistics of the Hungarian Academy of Sciences.

[2] Alan Munn and Cristina Schmitt (2005). Number and indefinites. Lingua, 115(6), 821-855.

[3] Anthony Kroch and Ann Taylor (2000). The Penn-Helsinki Parsed Corpus of Middle English (PPCME2). Department of Linguistics, University of Pennsylvania. CD-ROM, second edition, (http://www.ling.upenn.edu/hist-corpora/).

[4] Anthony Kroch, Beatrice Santorini, and Lauren Delfs (2004). The Penn-Helsinki Parsed Corpus of Early Modern English (PPCEME). Department of Linguistics, University of Pennsylvania. CD-ROM, first edition, (http://www.ling.upenn.edu/hist-corpora/).

[5] Ann Taylor, Arja Nurmi, Anthony Warner, Susan Pintzuk, and Terttu Nevalainen (2006). The York-Helsinki Parsed Corpus of Early English Correspondence (PCEEC). Department of Linguistics, University of York. Oxford Text Archive, first edition, (http://www-users.york.ac.uk/~lang22/PCEEC-manual/index.htm).

[6] Beth Randall (2010). CorpusSearch (http://corpussearch.sourceforge.net/index.html).