

Corpus Linguistics (L615)

Application #3: Language Learning

Markus Dickinson

Department of Linguistics, Indiana University
Spring 2009

Corpora are useful for language teaching and language learning:

- ▶ Direct use of corpora: teaching students to exploit corpora for learning
 - ▶ e.g., <http://vlc.polyu.edu.hk/concordance/>
- ▶ Indirect use of corpora: reference publishing, materials development, language testing
- ▶ Teaching-oriented corpus development: LSP corpora, L1 developmental corpora, L2 learner corpora
 - ▶ These corpora affect what is taught and how it is taught

Corpora in general allow for more exploration:
“illustration-interaction-induction”

Sinclair & Renouf (1988) proposed a *lexical syllabus*, organized around the notion of lexis and focusing on:

- a) 'the commonest word forms in a language'
- b) 'the central patterns of usage'
- c) 'the combinations which they usually form'

Corpora can enhance language testing by being used:

- ▶ to archive examination scripts
- ▶ to develop test materials
- ▶ to optimize test procedures
- ▶ to improve the quality of test making
- ▶ to validate tests
- ▶ to standardize tests

Rare but salient features?

One concern about the corpus-based approach to language teaching is that rare features are often ignored

- ▶ Yet these could be useful for learners to know, perhaps even more salient

Corpus data is often also decontextualized

These problems are fairly easily addressed, but must be noted

- ▶ Concordances of examples can show rare/atypical examples
- ▶ A classroom setting can provide some contextualization

Enriching reality: language corpora in language pedagogy

Gavioli & Aston 2001

Corpus Linguistics

Application #3:
Language Learning

Language learning

Gavioli & Aston

Thurstun & Candlin

Grammatical
morphemes

Can corpora capture reality?

- ▶ Test claims based on intuition that are passed on to students
 - ▶ Does *real* have positive associations?
 - ▶ Frequent corpus examples: *real life, the real world, real problems*
- ▶ Question whether low-frequency terms are important to teach
 - ▶ e.g., *double-Dutch* rarely occurs
- ▶ Question whether high-frequency terms are left out for a reason
 - ▶ e.g., Should *tend to* be taught if it appears as often as *ought*?
 - ▶ Difficulty is a factor, too, of course (e.g., “That’s enough, *don’t you think?*”)

Concordancing & the teaching of vocabulary of academic English

Thurstun & Candlin 1998

Corpus Linguistics

Application #3:
Language Learning

Language learning

Gavioli & Aston

Thurstun & Candlin

Grammatical
morphemes

Focus on learning lexical items that are common to students across various disciplines, i.e., “academic vocabulary”

- ▶ Focus on a restricted set of vocabulary, namely particular rhetorical functions
- ▶ Use “concordancing techniques to provide the student with intensive exposure to the uses of these items”

Learning these words better will help students write better academic papers

- ▶ Started with University Word List (Nation 1990)
- ▶ Selected a small set of words for a variety of rhetorical functions
 - ▶ Used frequency of use and “our own perception” to determine the word list

Why only about 150 items?

- ▶ Deal in detail with selected items:
 - ▶ students are exposed to “multiple examples of the same vocabulary item in context”
- ▶ Using a concordancer in this way can help make students aware of collocates
 - ▶ This is a way to help students develop an ability to guess the meaning of unknown words in context,
 - ▶ in addition to learning how to use the 150 keywords

The rhetorical functions

Some examples, with some keywords:

- ▶ Stating the topic of your writing
 - ▶ factor, issue, concept
- ▶ Referring to the research literature
 - ▶ evidence, research, source
- ▶ Reporting the research of others
 - ▶ according to, suggest, claim

The keywords appear frequently (at least once every 6000 words in a corpus)

- ▶ *unlikely* and *summary* are not used as frequently, but still kept
 - ▶ These are still useful for dealing with modality & creating final statements, respectively

Are there cases which are cause for revising what is taught?

- ▶ e.g., students are told not to pluralize *researches*
 - ▶ But this appears 10 times in the corpus, “indicating that it is, in fact, accepted practice in published texts”
 - ▶ Is that enough to change the way the (non-)plural of *research* is presented?
- ▶ e.g., reporting verbs appear in the present tense, but students are told to use the simple past

L2 acquisition of grammatical morphemes

Analyzing the L2 production of a learner can help better understand the L2 acquisition process.

Some ways to describe learner language (Ellis 1994):

- ▶ study of learner errors
 - ▶ Helped to develop the idea of an interlanguage for learners' constructed mental grammars
- ▶ study of developmental patterns
 - ▶ Dulay and Burt (1973) studied the acquisition order of grammatical features
 - ▶ Found this to be rather systematic
- ▶ study of variability
- ▶ study of pragmatic features

Learner corpora provide information relevant to these studies

Most useful if they are annotated with:

- ▶ properties about learner misuse
- ▶ properties about general grammatical patterns
- ▶ learner properties
 - ▶ Longitudinal learner corpora would be most useful for studying acquisition patterns
 - ▶ Cross-sectional corpora can still provide some insights

We'll use the International Corpus of Learner English (ICLE) for our studies (advanced learners)

Morpheme studies

Studies about morpheme acquisition order have shown this order to be preferred:

| Order | Morpheme | Example |
|-------|-------------------------|----------------------|
| 1 | plural -s | books |
| 2 | progressive <i>-ing</i> | John is <i>going</i> |
| 3 | copula BE | John <i>is</i> here |
| 4 | auxiliary BE | John <i>is</i> going |
| 5 | articles | <i>the</i> books |
| 6 | irregular past tense | John <i>went</i> |
| 7 | third person -s | John likes books |
| 8 | possessive -s | John's book |

Caveat: doesn't distinguish a 1% difference between levels from a 25% difference

Problem-oriented corpus annotation

Corpus Linguistics

Application #3:
Language Learning

Language learning

Gavioli & Aston

Thurstun & Candlin

Grammatical
morphemes

Where we're going:

- ▶ Convert corpus header information into more suitable version
- ▶ POS tag the corpus
- ▶ Manually tag morphological errors
- ▶ Obtain accuracy rates of errors

Basic formatting

Let's use MMAX2 as our corpus annotation tool, to help us add POS and error annotation

- ▶ After that, we can convert the corpus to something else, if we want to use a better search tool or the like

The first question is: What format is the corpus in, and how can we get it into MMAX2 format?

- ▶ Since the ICLE files are basically text-only, they are fairly straightforward to load.

Download MMAX2 onto the machine you're using, if you don't already have it

Loading a text file into MMAX2

1. Run `./startmmax.sh` (unix) or `startmmax.bat` (windows)
2. Tools → Project Wizard
 - 2.1 Text Input File: Pick an ICLE file and click on *Analyse File*
 - 2.2 Tokenization: click on *Tokenize*
 - 2.3 Markable level: Click on *Add level* for each level to be added
 - ▶ Make *word* level (source=WORD)
 - ▶ Make *pos* level (source=WORD)
 - ▶ Make *error* level (source=WORD)
 - 2.4 .MMAX Project: Pick a project path; you'll likely want `basedata`, `scheme`, etc. as daughter directories of this path.

Changing the error annotation scheme

Change the `error_scheme.xml` file to these contents:

```
<?xml version="1.0" encoding="UTF-8"?>
<annotationscheme>
<attribute id="error_level" name="error" type="freetext">
  <value name="error"/>
</attribute>
</annotationscheme>
```

Changing the POS annotation scheme

Likewise, change the `pos_scheme.xml` file to these contents:

```
<?xml version="1.0" encoding="UTF-8"?>
<annotationscheme>
<attribute id="tag_level" name="tag" type="freetext">
  <value name="tag"/>
</attribute>
</annotationscheme>
```

Adding error tags

After changing those scheme files:

- ▶ Click on File → Load, to reload the project
- ▶ Select the .mmax file which was created by the Project Wizard

Now, we can add POS & error annotation to every word

1. A very nice feature is that we can actually create error annotation for strings of words
2. For example (using BGSU1001):
 - ▶ Click on “time” and select the error level
 - ▶ In the Markable window, you’ll see an “error” attribute that you can fill in: fill in something like “weird introduction”
 - ▶ Now, drag over the previous “It is” and select “Add to this markable”
 - ▶ This now means that all 3 words make up the span of this error tag

Getting POS tags inputted

We're going to skip some necessary steps, that we'll come back to later in the semester:

1. Convert corpus file to a POS tagger's format
 - ▶ Use the MMAX2 basedata file as a starting point, to make sure you have the right tokens (onewordperline.pl)
2. POS tag the corpus
3. Convert the output into an MMAX-compatible markable file (cf. BGSU1001_pos_level.xml, convert.pl)

This will give us POS tags, which we can then hand-clean, if need be

Error tagging from POS tags

Because we're interested in whether particular morphemes are used correctly, we can start the search by looking at POS tags

- ▶ To annotate different kinds of errors, we would need to go through the corpus, sentence-by-sentence

List of morpheme tags

| Morpheme | Example | |
|----------------------|---------|------------|
| article | <ART> | <ER_ART> |
| possessive -s | <POS> | <ER_POS> |
| third person -s | <3PS> | <ER_3PS> |
| irregular past tense | <IRPST> | <ER_IRPST> |
| auxiliary BE | <AUXBE> | <ER_AUXBE> |
| plural -s | <PL> | <ER_PL> |
| copula BE | <COP> | <ER_COP> |
| progressive -ing | <PROG> | <ER_PROG> |

Transforming tags to desired morpheme tags

I wrote a short Perl program (`ttransform.pl`) which tags Penn Treebank tags and generates the morpheme tags we want

A lot of lines look like this:

- ▶ `s/\bVBG\b/PROG/g;`
- ▶ i.e., transform all instances of VBG to PROG
 - ▶ For more robustness, I would rewrite the program, to first extract the tag and then only change tags

With this output, we can insert POS tags into the markable file

Manually error tagging

If we only want to annotate our new tags, one way is to first search for the right ones.

1. Tools → Query Console
2. You can enter a search query or load a query
 - ▶ See the `mmax2query.pdf` file for more information

For tonight, let's simply go through a small ICLE file to add annotation.

General results

If we have a large number of files, we can randomly sample the different kinds of tags we're interested in

- ▶ Then, error annotate those corpus positions

You can see the results for longitudinal data on p. 261 of the book

- ▶ They show that some of the acquisition patterns don't seem to hold across different data