Corpus Linguistics
(L615)
Application #3: Language Learning

Markus Dickinson
Department of Linguistics, Indiana University
Spring 2013
Corpora are useful for language teaching & 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 features could be useful for learners to know, perhaps even more salient in some cases

Corpus data is often also decontextualized

These problems can be easily addressed, but must be noted

▶ Concordances of examples can show rare/atypical examples

▶ A classroom setting can provide contextualization
Corpora can help capture reality by being able to:

- test intuition-based claims passed on to students
  - Does *real* have positive associations?
  - Frequent corpus examples: *real life, the real world, real problems*
- question if low-frequency terms are important to teach
  - e.g., *double-Dutch* rarely occurs
- question why high-frequency terms are left out
  - Should *tend* to be taught if it appears as often as *ought*?
  - Difficulty is a factor (e.g., “That’s enough, *don’t you think*?”)
Concordancing & the teaching of vocabulary of academic English
Thurstun & Candlin 1998

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
Methodology

- 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 leaning 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)

▶ /Volumes/Data/en/icle/ on miller
Studies about morpheme acquisition order have shown this order to be preferred:

<table>
<thead>
<tr>
<th>Order</th>
<th>Morpheme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>plural -s</td>
<td>books</td>
</tr>
<tr>
<td>2</td>
<td>progressive -ing</td>
<td>John is going</td>
</tr>
<tr>
<td>3</td>
<td>copula be</td>
<td>John is here</td>
</tr>
<tr>
<td>4</td>
<td>auxiliary be</td>
<td>John is going</td>
</tr>
<tr>
<td>5</td>
<td>articles</td>
<td>the books</td>
</tr>
<tr>
<td>6</td>
<td>irregular past tense</td>
<td>John went</td>
</tr>
<tr>
<td>7</td>
<td>third person -s</td>
<td>John likes books</td>
</tr>
<tr>
<td>8</td>
<td>possessive -s</td>
<td>John’s book</td>
</tr>
</tbody>
</table>

Caveat: doesn’t distinguish a 1% difference between levels from a 25% difference
Problem-oriented corpus annotation

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
Let’s use EXMARaLDA 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 EXMARaLDA format?

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

▶ But how can we add, e.g., automatic POS annotation?
Loading a text into EXMARaLDA

As you know, you can:

▶ Import a file
▶ Add a POS tier
▶ Add an error annotation tier
▶ Annotate by hand

But we want to:

▶ Automatically tokenize
▶ Automatically POS tag
▶ Put this into EXMARaLDA

Solution: put into scripts!
1. Tokenization

You can write your own tokenization script in Perl

- Or use NLTK’s or NSP’s or, as we’ll do, a tokenizer that comes with a POS tagger
2. Automatic POS tagging

TreeTagger

We’ll use the TreeTagger for tokenization & POS tagging

▶ http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

Be sure to follow the installation instructions carefully

▶ e.g., put all files into one directory before running, including the English parameter files

▶ I also needed to change permissions after installing

> sudo chown -R md7 *
> sudo chgrp -R staff *

▶ I also added the executable files to my path: in my .profile, I added:

export PATH=$PATH:/Users/md7/research/tools/treetagger/cmd:
/Users/md7/research/tools/treetagger/bin
Automatic POS tagging

How to run the tagger:

> tree-tagger-english BGSU1001.txt > BGSU1001.tts
reading parameters ...  
tagging ...  
  finished.  
> more BGSU1001.tts
It   PP    it
is   VBZ   be
time NN    time
,    ,     ,
that IN    that
our PP$   our
society NN   society
is   VBZ   be
### 3. POS conversion

**List of morpheme tags**

<table>
<thead>
<tr>
<th>Morpheme</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>article</td>
<td><code>&lt;ART&gt;</code></td>
</tr>
<tr>
<td>possessive -s</td>
<td><code>&lt;POS&gt;</code></td>
</tr>
<tr>
<td>third person -s</td>
<td><code>&lt;3PS&gt;</code></td>
</tr>
<tr>
<td>irregular past tense</td>
<td><code>&lt;IRPST&gt;</code></td>
</tr>
<tr>
<td>auxiliary <strong>be</strong></td>
<td><code>&lt;AUXBE&gt;</code></td>
</tr>
<tr>
<td>plural -s</td>
<td><code>&lt;PL&gt;</code></td>
</tr>
<tr>
<td>copula <strong>be</strong></td>
<td><code>&lt;COP&gt;</code></td>
</tr>
<tr>
<td>progressive -<em>ing</em></td>
<td><code>&lt;PROG&gt;</code></td>
</tr>
</tbody>
</table>
Transforming tags to desired morpheme tags

I wrote a short Perl program (transform.pl) which tags PTB 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, we should rewrite this, to first extract the tag and only change tags (not words)

Call it like so:

```
> cat BGSU1001.tts | perl ..:/scripts/transform.pl \n  > BGSU1001.new
```

(NB: the \ refers to continuing the command on one line)
There is a script to take TreeTagger output and convert it to EXMARaLDA’s XML format

- You can treat `merge.py` more or less as a black box today
- It requires `template.xml`, which is worth going through:
  - Note how `[TOKENS]`, `[LEMMA]`, and `[POS]` are placeholders for the data
  - The other layers have no starting values

Call this as:

```
> python merge.py -c BGSU1001.new -t template.xml -o BGSU1001.exb
```

(Thanks to Adriane Boyd for the basic template script!)
## EXMARaLDA interface

<table>
<thead>
<tr>
<th>Token</th>
<th>It</th>
<th>is</th>
<th>time</th>
<th>,</th>
<th>that</th>
<th>our</th>
<th>society</th>
<th>is</th>
<th>dominated</th>
<th>by</th>
<th>industrialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemma</td>
<td>it</td>
<td>be</td>
<td>time</td>
<td>,</td>
<td>that</td>
<td>our</td>
<td>society</td>
<td>be</td>
<td>dominate</td>
<td>by</td>
<td>industrialization</td>
</tr>
<tr>
<td>POS</td>
<td>PP</td>
<td>3PR</td>
<td>NN</td>
<td>,</td>
<td>IN</td>
<td>PPS</td>
<td>NN</td>
<td>3PR</td>
<td>VBN</td>
<td>IN</td>
<td>NN</td>
</tr>
<tr>
<td>Error Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SENT</td>
</tr>
<tr>
<td>Correction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Error tagging from POS tags

Because we’re interested in whether particular morphemes are used correctly, we start by looking at POS tags

- To annotate different kinds of errors, we would need to go through the corpus, sentence-by-sentence
- Let’s annotate by hand for a bit ...
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.