“Grammar” versus observed data
What is linguistics all about?

• In grammar but not observed:
  Arguably the mainstream focus of linguistic research
  – Why? Productivity, competence, etc
  – How? Introspection, experiments, etc

• Observed but not in grammar (?):
  The noisy part of language
  (slips of tongue, “I heard it but I’d never say that”, etc)

A bottom-up approach
But grammar is ultimately based on the observed data.

A strongly empirical view of linguistic research:

• Focus on what is observed.
• Where are the data? There’s no shortage...

⇒ big data research!

**Big data for linguistic research**
...as the theme of the 2015 Linguistic Summer Institute at UChicago:

https://lsa2015.uchicago.edu/

**Big data + Cantonese?**
Some (accessible) Cantonese corpora, by year of publication:

• The Hong Kong Cantonese Adult Language Corpus (Leung and Law 2001; Leung et al. 2004; Fung and Law 2013)

• Hong Kong Cantonese Child Language Corpus (Lee and Wong 1998)

• Cantonese Radio Corpus (Francis and Matthews 2005, 2006)

• The Hong Kong Bilingual Child Language Corpus (Yip and Matthews 2007)

• Early Cantonese Tagged Database (Yiu 2012)

• A Linguistic Corpus of Mid-20th Century Hong Kong Cantonese (Chin 2013)

• PolyU Corpus of Spoken Chinese (Yap et al. 2014)

• Hong Kong Cantonese Corpus (Luke and Wong 2015)
Big data + Cantonese?
To what extent are these resources usable and extensible for the general research community?

Issues:

• inconsistent/ad hoc data formats
• no general toolkits for handling data

PyCantonese
PyCantonese is a toolkit for handling Cantonese corpus data.

• Evolving and expanding
• It is a Python library – why Python?
  – a general-purpose programming language
  – the lingua franca for computational linguistics and natural language processing
• Similar data structures as in NLTK (Bird et al. 2009)
• An open-source tool
• Current collaborators: Litong Chen, Tsz-Him Tsui
• Full documentation (with installation instructions): http://pycantonese.org/

Accessing corpus data in PyCantonese
PyCantonese comes with builtin corpus data!
Currently, KK Luke’s HKCanCor is included.

The corpus provides word-segmented data with:

• characters
• part-of-speech tags
• Jyutping romanization
Accessing corpus data through PyCantonese

```python
>>> import pycantonese as pc
>>> corpus = pc.hkcancor
>>> corpus.number_of_words()
160956
>>> corpus.number_of_characters()
210567
```

Parsing Jyutping

Jyutping → onset, nucleus, coda, tone

```python
>>> import pycantonese as pc
>>> pc.jyutping('hou2')
[('h', 'o', 'u', '2')]
>>> pc.jyutping('hoeng1gong2')
[('h', 'oe', 'ng', '1'), ('g', 'o', 'ng', '2')]
```

Also provided: Conversion from Jyutping to Yale or to \texttt{LaTeX TIPA}

Basic search capabilities

Possible search queries depend heavily on what is encoded and annotated in the corpus data:

- Jyutping elements?
- Part-of-speech tags?
- Characters?

More here: http://pycantonese.org/searches.html
More search examples
Use filtering strategies for more complicated search queries.

*Example:* Find in HKCanCor all verb+noun word pairs.
(defined as 1st word tag = “V” and 2nd word tag == “N”)

*Approach:*

1. Find all words tagged as “V” together with the immediately following word.
2. Within the results from step 1, retain only cases where the second word has the tag of “N”.

Finding V+N word pairs

```python
>>> import pycantonese as pc
>>> corpus = pc.hkcancor
>>> v = pc.search(corpus, pos="V", word_right=1)
>>> len(v) # number of words with "V"
25364
>>> vn = list()
>>> for wordpair in v:
...   if not wordpair or len(wordpair) < 2:
...     continue
...   if wordpair[1][1] and wordpair[1][1] == "N":
...     vn.append(wordpair) # save V+N
>>> len(vn) # number of V+N word pairs
1535
```

Some V+N pairs found

```python
>>> for i in range(3):
...   print(vn[i])

[['聽_teng1', 'V'), ('朋友_pang4jau5', 'N')]  
[['跟_gan1', 'V'), ('旅行社_leoi5hang4se5', 'N')]  
[['搭_daap3', 'V'), ('飛機_fei1gei1', 'N')]
```

TODO: Allow regular expressions for search criteria.

e.g., the part-of-speech tag of interest could be anything in the tagset that begins with a “V” (= some sort of verb).
Recurrent problem: Part-of-speech tagging

Some issues of part-of-speech tagging:

1. How many tags do we use?
   - HKCanCor: 46 + tags
   - Google universal tagset: 12 tags (Petrov et al. 2011)

2. Relatedly, how fine-grained are the tags?
   - e.g., distinguish proper nouns and common nouns?

3. Human annotation work is time-consuming and costly.

魔鬼 gwai2 ‘ghost’

Examples from HKCanCor:
1. 好 hou2/D 鬼 gwai2/D1 細 sai3/A
   “very GWAI small”
2. 有 jau5/V1 鬼 gwai2/D1 今日 gam1jat6/T
   “resulting-in GWAI today”

What is the tag D1?
These two instances of gwai2 are very different.
(An expressive + negator in (2); see Beltrama and Lee (2015))

Current work:
Mapping HKCanCor to the universal PoS tagset by Petrov et al

A related issue: Word segmentation

Issues of word segmentation:

1. AB
   → compound or two separate words?

2. grammatical characters (e.g., aspect markers)
   → a separate word itself or part of a word?

Interrogatives A-not-A, A-not-AB

If we treat A-not-AB as three words...

What is hap in hap-m-happy?
Similarly, 鍾唔鍾意 “like or not”, etc.
(In HKCanCor, the first A is treated as an abbreviation, with a tag starting with “J”.)
Or perhaps things like A-not-AB should be treated as one word? (Lee 2012)

Same problem: aspect markers
Ongoing work

- Corpus data prep
  (The Leung-Law-Fung HKCAC, the Francis-Matthews CRCorpus)
- General tools thus derived

Comparing some Hong Kong Cantonese corpora
Both standard and non-standard data formats have been used.

Potential new tools in PyCantonese
...and a call for arms!

- A part-of-speech tagger
- Conversion between Jyutping and characters, both directions
  (Issues: Homophony and homography)
- Word segmentation (with all the usual problems!)

Data, data, data
Ultimately, what is observed is the data.

Data format:
• General direction for PyCantonese:
  Adopting the CHILDES CHAT format (MacWhinney 2000)

• Reasons:
  – Rich annotations
  – It is well documented and supported.
  – XML format available by conversion
    ⇒ readable by NLTK – and PyCantonese!

• What about non-conversational data?

Data prep:
• Other (publicly available) datasets out there?
• Audio(-visual) data?
• What annotations are desirable?

[Update 2015-09-22]

Additional notes and code snippets are available here:

References


Yap, Foong Ha, Ying Yang and Tak-Sum Wong. 2014. On the development of sentence final particles (and utterance tags) in Chinese. In Kate Beeching and Ulrich Detges (eds.), *Discourse functions at the left and right periphery*, 179-220. Leiden: Koninklijke Brill NV.
