Parsing and interrogating the Royal Society Corpus

Daniel McDonald

6th October 2016

Introduction

I've been parsing and interrogating the RSC. This means:

1. Turning the `jstor bundle` into something that is parseable, but with retrievable metadata
2. Experimenting with parsers
3. Parsing everything, reintroduce metadata to parser output
4. Developing tools: command interpreter, symbolic subcorpus management, automatic annotation, search filters
5. Exploring the parsed corpus via lexicogrammatical querying
6. (Trying to) generate some entropy/surprisal scores and methods

Preparing the corpus for parsing

We can't just parse the XML right away, and we don't want to just strip out all that high-quality metadata. So:

1. Make a version with and without metadata tags
2. Add some metadata along the way using `langdetect`:
   - Probable language
   - Probability that text is English
3. Parse the version without metadata
4. Use character offsets to add metadata to parser output

The script for this is very quick to run, and can be improved or extended for future efforts.

Things parsers don't like

1. Language/spelling change: noun/proper noun issues, *hath*, etc.
2. Text/metadata distinction
3. Figures and tables as sentences
4. Maths in text
5. Non-English text

Issue: text/metadata distinction

IV. Part of a Letter from Mr James Yonge to Mr John Haughton, F. R. S. concerning the internal use of Cantharides. Plymouth, July 17. 1702. SIR, A Gentlewoman of 54 years old, who for a long time had been tormented with frequent Fits of the Stone

Issue: figures, tables, references

13. Filix scandens Malab. pinnis integris alternatim sitis. 13 filix scandens Indica, ramulis ex adverso binis, foliis alternatim sitis, oblongis, angustis cuspidatis Ray V. 3. l. 3. p. 90. The top Leaf is often fork'd, the rest single. I have received it not only from fort St George, but also from the Grain and Gold Coasts of Guiney. 14. The Male Bangue. 14 Bange Clus. Exot 238. c. 25. & 290. c. 54. Fragos. 58. c. 26. Bangue arbor Cannabi similis ad omnia fere utilis seu Amsion (s. Opium) Linschot Ind. Or. pt. 4. c. 35. Bangue Cannabi simile I. B. Vol. 3. l. 30. p. 449. c. 71. Cannabi similis exotica C. B. 320. 4. C. B. phyt. 640.

Issue: maths

On the same principle we must proceed if such forms as cos (n log), sin (n log X), &c. are found in the second member. Ex. 3. Given s2 @ +d u +-=log ( ). Putting x=-g, we have D2u+eu=O. Make u=A+Bd, then on reducing D2A+2DB+ A+-(D2B+s-B-I)-0, whence, as in preceding examples, D2A+2DB+A = 0, D2B+-B= 1. This system of equations differs from those before considered, in that the second members do not both vanish. The fundamental theorem gives A==:am;, B=Zbm;, 1{ (mza+2mbmtI,)m } >2{ (mZbm +b, _O'}= --l,1 6237whence m2aq+2mbm+aa_.l=O for all values of m, and m2bm+-b_-l=O for all values of m except m=0, which gives m2bm+-b _ I, or b-=l; also from the other equation, a_-=O. From these, the values of am, bm, corresponding to negative values only of n, may be determined; whence writing x for s, and solving the above equations relatively to a,_and b^_,, we have a*2 a,3 u=5a_+ a; a +&C. ++^+ + &c. +logx (b,+? + &C.) where a_=0, b=-i1; and in general a_ (m2am+2mb.), b_I= -m2bm.

Issue: Non-English

```
Verum priusquam ulterius progrediar hoc te monitum velim me usurpare
illa quae demonstravit Clarissimus Newtonus in pag. 251, 252 & 253
Princ. Phil. circa momentanea incrementa vel decrementa quantitatum quae
fluxu continuo crescunt vel decrescunt, praesertim quod dignitatis
cujuscunque A n/m momentum sit n/ma A n/m 1. Porro data fluxione n/m aA n
/m 1 vicissim reperiri potest quantitas fluens A n/m, 10 tollendo a de
fluxione,20 fluxionis Indicem unitate augendo, 30 denique fluxionem
dividendo per Indicem sic unitate auctum. Curvae abscissa designabitur
deinceps per x, ejus fluxio per x, ordinatim applicata per y, ejusque
fluxio per y His positis ut ad quadraturas deveniamus, 10 assumatur
valor ordinatim applicatae ope aequationis naturam Curvae exprimentis.
20 Multiplicetur hic valor
```

Solutions

Ways to solve these problems for the investigation of the RSC:

1. Via pre-processing:
   - Searching, manual/semi automatic correction
   - Time consuming and expensive, but can create high quality resources
2. Via post-processing:
   - Use language identification metadata to filter corpus during search
   - Exclude based on lexicogrammatical criteria
   - Fast, but sacrifices data
3. Iterative development: back and forth between the two (yes!)

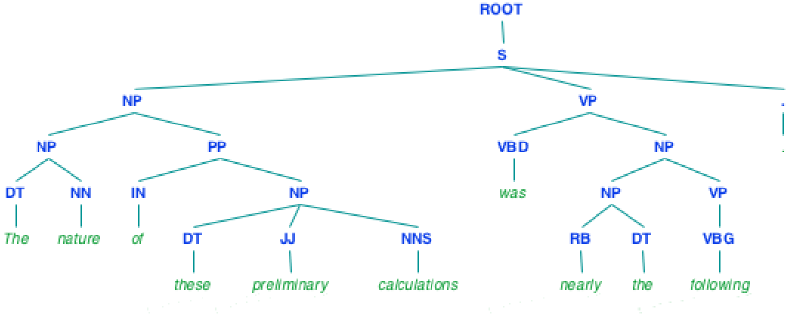
Parsing possibilities

- A lot of messy data, so speed and accuracy are both issues
- Faster parsers provide less, and are less accurate
- `spaCy`: no constituency, no coreference resolution, no `copula=head`
- `CoreNLP`: slower, Java trickiness
- The ideal solution would include:
  - Cleaner data
  - Using the best POS tagging info in the model
  - training a model from corrected parser output
  - training multiple models for different time periods

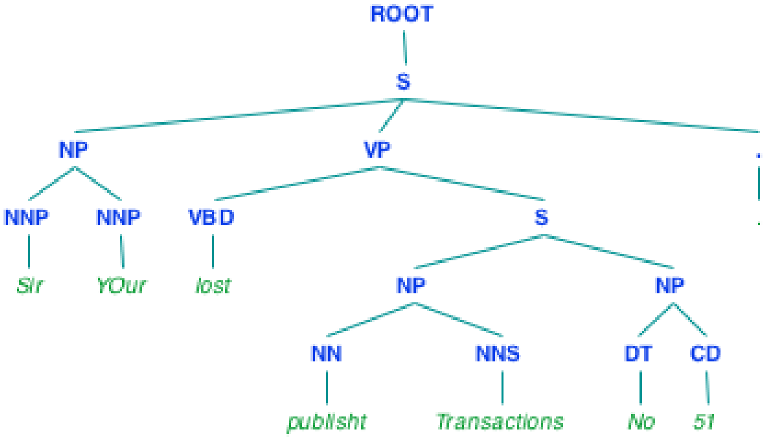
Parsing

1. Using `CoreNLP` (via `corpkit`)
   - Provides lemmatisation, POS, NER, constituency, typed dependency, coreference resolution
   - Coreference resolution is slow, memory intensive, inaccurate, causes a few big errors
2. Great example of an *embarrassingly parallel* job
3. Parsing time unknown—a few days with 15 processes, some Java errors

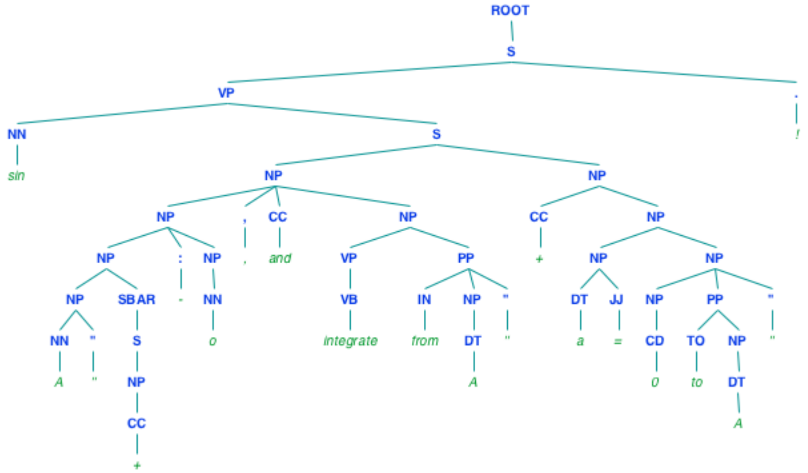
The good


ROOT
S
NP
VP
.
NP
PP
VBD
NP
.
DT
NN
IN
NP
was
NP
VP
The
nature
of
DT
JJ
NNS
RB
DT
VBG
these
preliminary
calculations
nearly
the
following

The bad


ROOT
S
NP
VP
.
NNP
NNP
VBD
S
.
Sir
YOur
lost
NP
NP
NN
NNS
DT
CD
publisht
Transactions
No
51

The ugly


ROOT
S
VP
.
NN
S
!
sin
NP
NP
NP
,
CC
NP
CC
NP
NP
:
NP
,
and
VP
PP
+
NP
NP
NP
SBAR
-
NN
VB
IN
NP
"
DT
JJ
NP
PP
"
NN
"
S
o
integrate
from
DT
"
a
=
CD
TO
NP
"
A
"
NP
A
0
to
DT
CC
A
+

corpkit: tool for analysis

corpkit is my module for making and querying parsed corpora.

1. Traverse and output grammatical annotations and metadata
2. Interrogate and concordance at the same time—use concordance to identify/remove false positives, do thematic categorisation, and then recalculate results
3. Language models including grammatical information (beta)
4. Fast, cross platform, open source, documented, version control
5. Interfaces: Python API, GUI, and more recent **CQP-style interpreter**

Tool extensions

Some feedback on `corpkit` led to:

1. Metadata as subcorpora and filter
2. Symbolic and multi-level subcorpora
3. Semi-automatic annotation via concordancing
4. Fully scriptable

corpkit interpreter demo

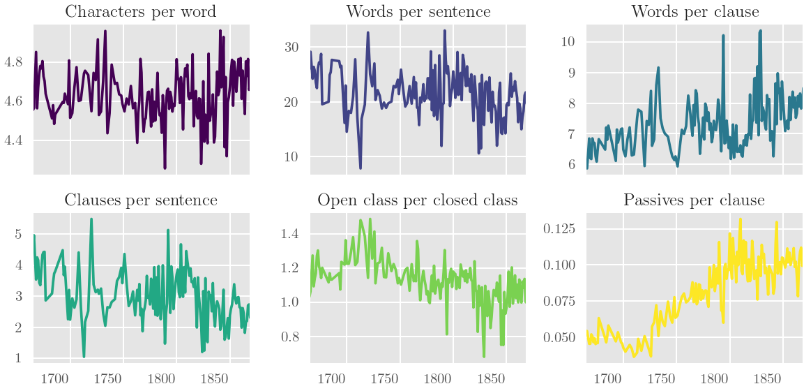
http://corpkit.readthedocs.io

RSC: interrogation

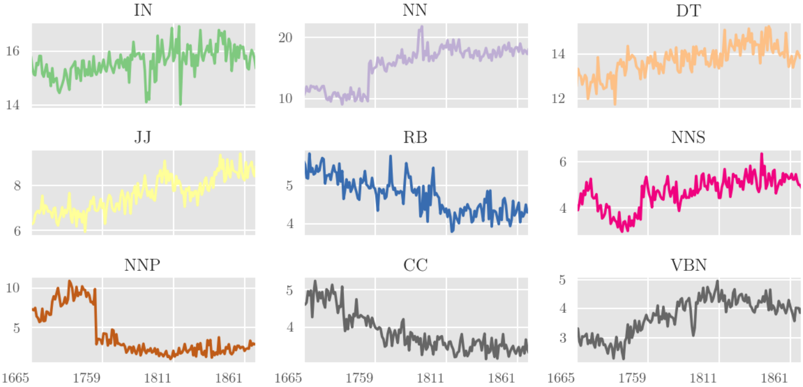
The strategy:

- Use years as subcorpora
- Search for sentences whose root lemma is in VerbNet
- Tag these sentences and filter out the others
- Filter out texts (not sentences, yet) that are not English
- Get general features, and use these to generate relative frequencies for more specific phenomena
- Automate the process of discovering longitudinal change
- Keep everything logged, saved, and on GitHub

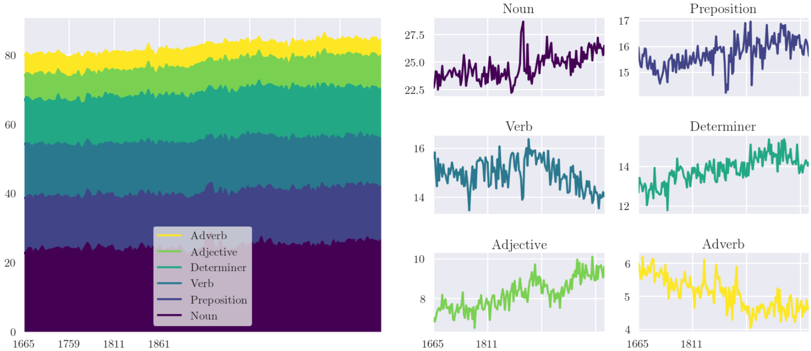
Derived features



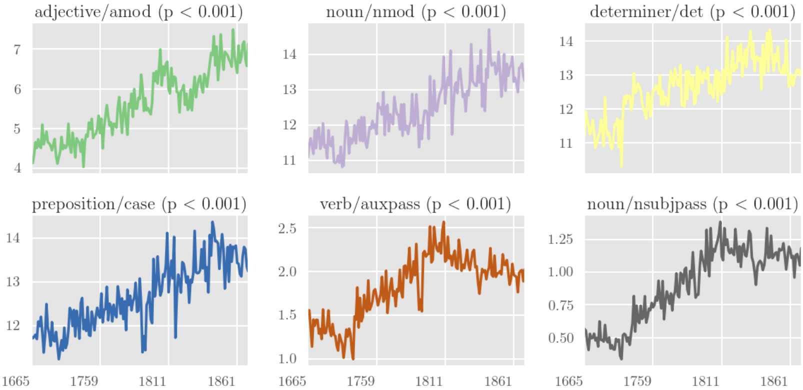
POS tag and parser accuracy



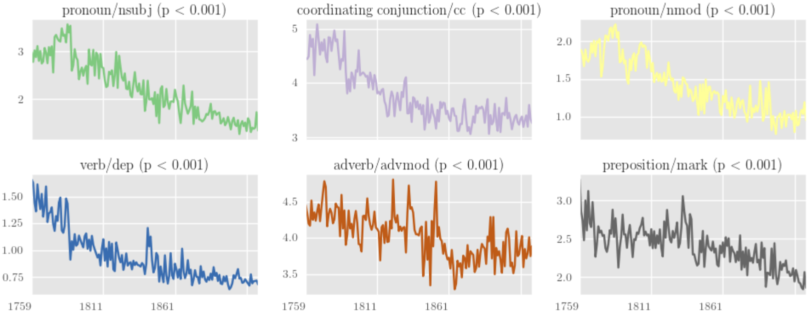
Wordclass and parser accuracy



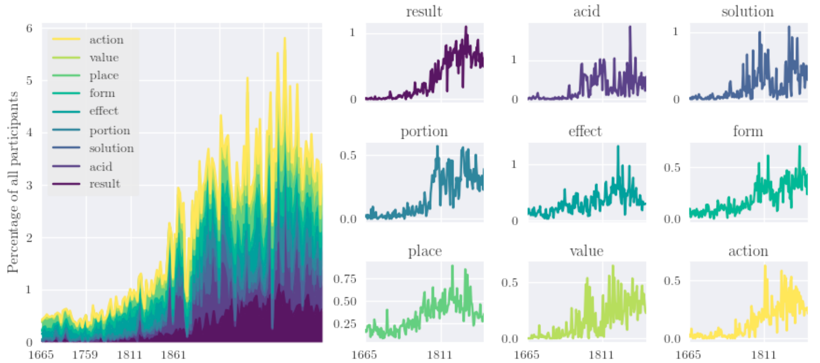
Mixing wordclass and dep type



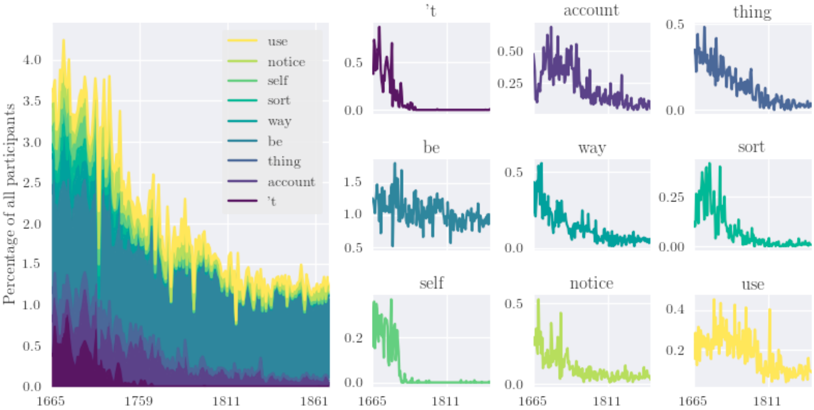
Mixing wordclass and dep type



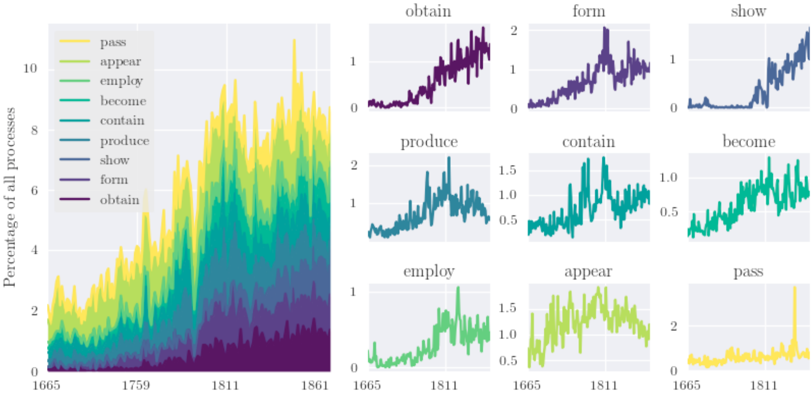
Increasingly frequent participants



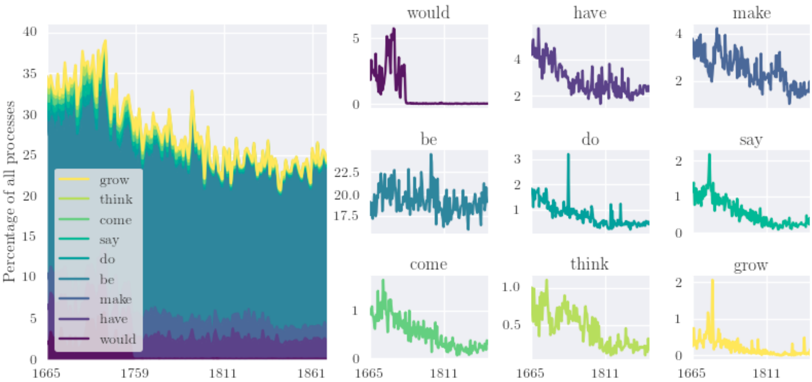
Decreasingly frequent participants



Increasingly frequent processes



Decreasingly frequent processes



Discussion

1. Parsing will need to be repeated reliably as the corpus is improved
2. To what extent will this approach model parser accuracy rather than language change?
3. Parser accuracy *as* language change

Next steps

1. Use the annotations to get bad text in the original bundle
2. Use tool for syntactic language models?
3. Measure parser accuracy
4. Word/sentence/text level annotations for entropy?
5. What should the training data be? The corpus? Subcorpus? Training set from (groups of) subcorpora?