

# IndVarBNC

March 30, 2020

## 1 Load packages

Package requirements are stored in `requirements.yml`.

```
[1]: import os
import random
from collections import defaultdict
from pprint import pprint

import numpy as np
import pandas as pd

from lxml import etree

import altair as alt
#alt.data_transformers.disable_max_rows()
```

## 2 Variables

BNC2014 needs to be downloaded for this script to work. It can be obtained from the official [BNC website](#).

The following variables need to be updated to the corpus' local path. In the current setting the BNC2014 data were stored in the project folder in the folder `data`, so relative paths were used.

```
[2]: dir_corpus = 'data/spoken/tagged/'
dir_meta = 'data/spoken/metadata/'
```

## 3 Load and parse XML

```
[3]: f_names = os.listdir(dir_corpus)
f_paths = [f"{dir_corpus}{f_name}" for f_name in f_names]
```

```
[4]: def get_xml(f_path):
      with open(f_path, 'r') as f:
          f = f.read()
          xml = etree.fromstring(f)
          return xml
```

## 4 Corpus size

### 4.1 Texts

Calculate the total number of texts in the corpus.

```
[5]: %%time
      texts = []
      for f_path in f_paths:
          xml = get_xml(f_path)
          id = xml.get('id')
          texts.append(id)
```

CPU times: user 40.6 s, sys: 6.72 s, total: 47.4 s  
Wall time: 49.5 s

```
[6]: print(f"number of documents in the corpus: {len(texts)}")
```

number of documents in the corpus: 1251

### 4.2 Speakers

1. Store all speakers in the corpus.
2. Store the total number of words each speaker has contributed to the corpus.

```
[7]: %%time
      speakers_words = defaultdict(int)
      for f_path in f_paths:
          xml = get_xml(f_path)
          for u in xml.iter('u'):
              speaker = u.get('who')
              n_words = len([w for w in u.iter('w')])
              speakers_words[speaker] += n_words
```

CPU times: user 47 s, sys: 6.69 s, total: 53.7 s  
Wall time: 56.6 s

### 4.2.1 Number of speakers

```
[8]: print(f"number of speakers: {len(speakers_words)}")
```

number of speakers: 671

### 4.2.2 Words per speaker

```
[9]: df_speakers_words = pd.DataFrame(list(speakers_words.items()),  
    ↪ columns=['speaker', 'n_words'])
```

```
[10]: df_speakers_words.sort_values('speaker', ascending=True, inplace=True)
```

```
[44]: df_speakers_words
```

```
[44]:
```

	speaker	n_words
468	S0001	3000
459	S0002	8535
440	S0003	1893
312	S0004	3634
143	S0005	1449
..	...	...
219	S0691	2135
214	S0692	1105
26	UNKFEMALE	28108
32	UNKMALE	30316
5	UNKMULTI	1375

[671 rows x 2 columns]

### 4.2.3 Write out

```
[12]: with open('out/metadata/speakers_words.csv', 'w') as f_out:  
    df_speakers_words.to_csv(f_out, index=False)
```

## 4.3 Words

```
[13]: %%time  
n_words = 0  
for f_path in f_paths:  
    xml = get_xml(f_path)  
    for w in xml.iter('w'):  
        n_words += 1
```

CPU times: user 44.5 s, sys: 6.42 s, total: 50.9 s  
Wall time: 53.8 s

```
[14]: print(f"total number of words in the corpus: {n_words}")
```

total number of words in the corpus: 11422615

## 5 Query for that's ADJ

```
[15]: blacklist = ['to', 'timing', 'news', 'bullshit', 'awesome', 'enough']
```

```
[16]: def get_thats_adj_start_disc(u):
    query = "that 's ADJ - start_disc"
    n_slots = 3
    hits = []
    words = [w for w in u.iter('w')]
    for w in words:
        # check first token
        if w.text.lower() == 'that' and w.get('pos') == 'DD1':
            # check sequence length to avoid crossing utterance boundaries
            if words.index(w) + n_slots - 1 < len(words):
                w2 = words[words.index(w) + 1]
                w3 = words[words.index(w) + 2]
                # check second token
                if w2.text.lower() == "'s":
                    # check third token
                    if w3.get('pos') == "JJ":
                        # store information about hits
                        hit = {}
                        hit['doc'] = xml.get('id')
                        hit['utterance'] = u.get('n')
                        hit['speaker'] = u.get('who')
                        hit['query'] = query
                        hit['result'] = ' '.join(
                            [
                                w.text,
                                w2.text,
                                w3.text,
                            ]
                        )
                        hit['adj_text'] = w3.text
                        hit['adj_semtag'] = w3.get('usas')
                        # check whether pattern occurs at start of utterances or
                        ↳ is only preceded by an interjection
                        if words.index(w) == 0 or (words[0].get('pos') == 'UH'
                        ↳ and words.index(w) == 1):
```

```

        # exclude hits that are followed by blacklist items
        if words.index(w) + n_slots < len(words):
            w4 = words[words.index(w) + n_slots]
            if w4.text.lower() not in blacklist:
                hits.append(hit)
        else:
            hits.append(hit)

    return hits

```

```

[17]: %%time
hits = []
for f_path in f_paths:
    xml = get_xml(f_path)
    for u in xml.iter('u'):
        hits.extend(get_thats_adj_start_disc(u))

```

CPU times: user 49.6 s, sys: 6.22 s, total: 55.8 s

Wall time: 59.9 s

```

[21]: print(f"number of hits: {len(hits)}")

```

number of hits: 8027

```

[22]: print('random selection of 10 hits:\n')
for hit in random.choices(hits, k=10):
    print(hit['result'])

```

random selection of 10 hits:

```

that 's great
that 's true
that 's right
that 's nice
that 's right
that 's fatal
that 's good
that 's fine
that 's nice
that 's awful

```

```

[23]: df_hits = pd.DataFrame(hits)
df_hits

```

```

[23]:
   doc utterance speaker      query      result \
0   SN64      521   S0588  that 's ADJ - start_disc  that 's good
1   SN64      577   S0588  that 's ADJ - start_disc  that 's strange
2   SN64      834   S0588  that 's ADJ - start_disc  that 's ridiculous

```

3	SN64	854	S0588	that 's ADJ - start_disc	that 's ridiculous
4	SN64	980	S0588	that 's ADJ - start_disc	that 's atrocious
...	...	...	...	...	...
8022	S37K	669	S0058	that 's ADJ - start_disc	that 's alright
8023	S37K	740	S0058	that 's ADJ - start_disc	that 's interesting
8024	S37K	1555	S0058	that 's ADJ - start_disc	that 's true
8025	SMHY	139	S0037	that 's ADJ - start_disc	that 's true
8026	SMHY	190	S0115	that 's ADJ - start_disc	that 's Well

	adj_text	adj_semtag
0	good	A5:1
1	strange	A6:2
2	ridiculous	S1:2:6
3	ridiculous	S1:2:6
4	atrocious	A5:1
...	...	...
8022	alright	A5:1
8023	interesting	X5:2
8024	true	A5:2
8025	true	A5:2
8026	Well	B2

[8027 rows x 7 columns]

## 6 Semantic category descriptions

BNC2014 data are tagged using the USAS semantic tagger. Information for each hit were stored in `df_hits` and are now merged with semantic category descriptions from the USAS tagset which was download from <http://ucrel.lancs.ac.uk/usas/>.

```
[24]: semtags = pd.read_csv(
        'semtags.csv',
        sep='\t',
        names=['tag', 'desc']
    )
```

```
[25]: semtags.tag = semtags.tag.str.replace('.', ':')
```

```
[26]: df_hits = pd.merge(df_hits, semtags, left_on='adj_semtag', right_on='tag',
        how='left')
```

```
[27]: df_hits.drop('tag', axis=1, inplace=True)
```

```
[28]: df_hits.rename(columns={'desc': 'adj_semdesc'}, inplace=True)
```

```
[29]: df_hits
```

```
[29]:
```

	doc	utterance	speaker	query	result \
0	SN64	521	S0588	that 's ADJ - start_disc	that 's good
1	SN64	577	S0588	that 's ADJ - start_disc	that 's strange
2	SN64	834	S0588	that 's ADJ - start_disc	that 's ridiculous
3	SN64	854	S0588	that 's ADJ - start_disc	that 's ridiculous
4	SN64	980	S0588	that 's ADJ - start_disc	that 's atrocious
...	...	...	...	...	...
8022	S37K	669	S0058	that 's ADJ - start_disc	that 's alright
8023	S37K	740	S0058	that 's ADJ - start_disc	that 's interesting
8024	S37K	1555	S0058	that 's ADJ - start_disc	that 's true
8025	SMHY	139	S0037	that 's ADJ - start_disc	that 's true
8026	SMHY	190	S0115	that 's ADJ - start_disc	that 's Well

  

	adj_text	adj_semtag	adj_semdesc
0	good	A5:1	Evaluation:- Good/bad
1	strange	A6:2	Comparing:- Usual/unusual
2	ridiculous	S1:2:6	Sensible
3	ridiculous	S1:2:6	Sensible
4	atrocious	A5:1	Evaluation:- Good/bad
...	...	...	...
8022	alright	A5:1	Evaluation:- Good/bad
8023	interesting	X5:2	Interest/boredom/excited/energetic
8024	true	A5:2	Evaluation:- True/false
8025	true	A5:2	Evaluation:- True/false
8026	Well	B2	Health and disease

[8027 rows x 8 columns]

```
[30]: query = df_hits.loc[0, 'query']
f_name_out = query.replace(" ", "-")
dir_hits_out = 'hits/'
```

```
[31]: with open(f"out/{dir_hits_out}{f_name_out}.csv", 'w') as f_out:
df_hits.to_csv(f_out, index=False)
```

## 7 Metadata

### 7.1 Speakers

```
[32]: head_speakers = pd.read_csv(
    f"{dir_meta}metadata-fields-speaker.txt",
    delimiter='\t',
    skiprows=1,
```

```

        index_col=0
    )

```

```

[33]: speakers = pd.read_csv(
        f"{dir_meta}bnc2014spoken-speakerdata.tsv",
        delimiter='\t',
        names=head_speakers['XML tag'],
        index_col=0
    )

```

```

[34]: speakers

```

```

[34]:
    exactage  age1994  agerange  gender  nat \
S0001      32    25_34    30_39      F  British
S0002     NaN  Unknown    19_29      F  British
S0003     NaN  Unknown    19_29      F  British
S0004     NaN  Unknown    30_39      M  British
S0005     NaN   60plus    80_89      F  British
...
S0691     45    45_59    40_49      F  British
S0692     22    15_24    19_29      M  British
UNKFEMALE     NaN  Unknown  Unknown      F      NaN
UNKMALE     NaN  Unknown  Unknown      M      NaN
UNKMULTI     NaN  Unknown  Unknown      X      NaN

    birthplace  birthcountry  l1  lingorig \
S0001      Wordsley, West Midlands  England  English  England
S0002      Birmingham  England  English  England
S0003  Royal Leamington Spa, Warwickshire  England  English  England
S0004      NaN  Germany  English  England
S0005      Birmingham  England  English  England
...
S0691      Barrow-In-Furness  UK  English  England
S0692      Barrow-in-Furness  England  English  England
UNKFEMALE      NaN  NaN  NaN  NaN
UNKMALE      NaN  NaN  NaN  NaN
UNKMULTI      NaN  NaN  NaN  NaN

    dialect_rep  ...  dialect_l2  dialect_l3  dialect_l4 \
S0001  None indicated  ...  unspecified  unspecified  unspecified
S0002  Midlands  ...  england  midlands  unspecified
S0003  Northern  ...  england  north  unspecified
S0004  Northern  ...  england  north  unspecified
S0005  Midlands  ...  england  midlands  unspecified
...
S0691  Northern/ Cumbrian  ...  england  north  unspecified
S0692  Northern  ...  england  north  unspecified

```



UNKFEMALE	None indicated	...	unspecified	unspecified	unspecified	
UNKMALE	None indicated	...	unspecified	unspecified	unspecified	
UNKMULTI	None indicated	...	unspecified	unspecified	unspecified	

  

	edqual	occupation	socgrade	nssec	12	\
S0001	5_postgrad	University researcher	A	1_2	NaN	
S0002	5_postgrad	Teacher	B	2	NaN	
S0003	4_graduate	Student	E	uncat	NaN	
S0004	5_postgrad	Engineer	C2	5	NaN	
S0005	2_secondary	Insurance Broker (retired)	E	8	NaN	
...	...	...	...	...	...	
S0691	3_sixthform	dental nurse (trainee)	D	6	NaN	
S0692	3_sixthform	Sales Assistant (Part time)	D	6	NaN	
UNKFEMALE	9_unknown	NaN	unknown	unknown	NaN	
UNKMALE	9_unknown	NaN	unknown	unknown	NaN	
UNKMULTI	9_unknown	NaN	unknown	unknown	NaN	

  

	fls	in_core
S0001	NaN	n
S0002	Japanese -- Intermediate	n
S0003	NaN	n
S0004	Spanish -- Beginner	n
S0005	French -- Beginner	n
...	...	...
S0691	NaN	y
S0692	NaN	n
UNKFEMALE	NaN	n
UNKMALE	NaN	n
UNKMULTI	NaN	n

[671 rows x 24 columns]

## 7.2 Texts

```
[35]: head_texts = pd.read_csv(
    f"{dir_meta}metadata-fields-text.txt",
    delimiter='\t',
    skiprows=1,
    index_col=0
)
```

```
[36]: texts = pd.read_csv(
    f"{dir_meta}bnc2014spoken-textdata.tsv",
    delimiter='\t',
    names=head_texts['XML tag'],
    index_col=0
)
```

)

[37]: texts

```
[37]:      rec_length  rec_date  rec_year  rec_period  n_speakers  \
S23A      1:50:43  2014-12-27      2014      2014_Q4          4
S24A      0:17:24  2014-09-12      2014      2014_Q3          2
S24D      0:20:00  2016-01-14      2016      2016_Q1          3
S24E      0:45:53  2015-09-15      2015      2015_Q3          3
S263      2:00:00  2016-02-07      2016      2016_Q1          4
...
SZVB      1:00:31  2015-11-02      2015      2015_Q4          2
SZVC      0:32:00  2015-09-14      2015      2015_Q3          2
SZW4      0:21:09  2015-10-19      2015      2015_Q4          2
SZXQ      0:40:44  2012-03-21      2012      2012_Q1          2
SZYV      0:21:20  2015-11-04      2015      2015_Q4          2

      list_speakers                                     rec_loc  \
S23A  S0021 S0032 S0094 S0095                               Speakers' home
S24A                               S0261 S0262      Modern Art Museum, London
S24D      S0653 S0654 S0655                               Home kitchen, Comberton
S24E      S0519 S0520 S0521                               Hunsonby, Cumbria
S263  S0588 S0589 S0590 S0616                               ANON's home
...
SZVB      S0517 S0525                               (ANON's home, Fradley, Staffs)
SZVC      S0324 S0325                               ANON's home, Linton
SZW4      S0509 S0510      ANON & ANON's home, Hastings
SZXQ      S0058 S0120      Botanic Gardens, Cambridge
SZYV      S0428 S0432  Cambridge University Press Printing House

      relationships  \
S23A  Close family, partners, very close friends
S24A  Close family, partners, very close friends
S24D  Close family, partners, very close friends
S24E  Close family, partners, very close friends
S263  Close family, partners, very close friends
...
SZVB  Close family, partners, very close friends
SZVC  Close family, partners, very close friends
SZW4  Close family, partners, very close friends
SZXQ      Friends, wider family circle
SZYV      Friends, wider family circle

      topics  \
S23A  Computer programming, food, wine, temperature,...
S24A      The art
S24D      Lego Ninjago, Minecraft worlds
```

S24E food, exercise, choir, family plans, family me...  
S263 NaN  
...  
SZVB Babies, family, friends  
SZVC school orchestra (windband), Playing the Clari...  
SZW4 Poetry, Morning Routine, Food, Social Events, ...  
SZXQ TV, languages, friends, holidays, offices, comedy  
SZYV Babies, moving house, sharing clothes, sibling...

activity \  
S23A Catching up with family over food and presents  
S24A A couple discussing modern art at a museum  
S24D Spending time on electronic toys instead of re...  
S24E Midweek family dinner  
S263 NaN  
...  
SZVB Sisters talking about their family (new baby d...  
SZVC Friends talking about school  
SZW4 Mother and Daughter  
SZXQ NaN  
SZYV Lunchtime chat

		conv_type	conventions	in_sample	\
S23A	Discussing, explaining, anecdote telling	Revised		n	
S24A	Discussing, explaining, inquiring	Revised		y	
S24D	Discusing, explaining	Revised		n	
S24E	Discussing, explaining, Inquiring, advising, a...	Revised		n	
S263	Discussing, explaining	Revised		n	
...	...	...	...		
SZVB	Discussing, explaining, inquiring, anecdote te...	Revised		n	
SZVC	discussing, explaining, inquiring, complaining...	Revised		n	
SZW4	Discussing, inquiring, anecdote telling	Revised		n	
SZXQ	Discussing, explaining, inquiring, complaining...	Original		y	
SZYV	Discussing, explaining, inquiring	Revised		n	

transcriber  
S23A T15  
S24A T09  
S24D T18  
S24E T09  
S263 T10  
...  
SZVB T15  
SZVC T10  
SZW4 T18  
SZXQ T11  
SZYV T19

[1251 rows x 14 columns]

### 7.3 Merge hits and metadata

```
[38]: df_merged = pd.merge(df_hits, speakers, left_on='speaker', right_on=speakers.  
      ↪index)
```

```
[39]: df_merged = pd.merge(df_merged, texts, left_on='doc', right_on=texts.index)
```

```
[46]: df_merged.head()
```

```
[46]:      doc utterance speaker      query      result \  
0  SN64      521   S0588  that 's ADJ - start_disc  that 's good  
1  SN64      577   S0588  that 's ADJ - start_disc  that 's strange  
2  SN64      834   S0588  that 's ADJ - start_disc  that 's ridiculous  
3  SN64      854   S0588  that 's ADJ - start_disc  that 's ridiculous  
4  SN64      980   S0588  that 's ADJ - start_disc  that 's atrocious
```

```
      adj_text adj_semtag      adj_semdesc exactage age1994 ... \  
0      good      A5:1      Evaluation:- Good/bad      49  45_59 ...  
1     strange      A6:2  Comparing:- Usual/unusual      49  45_59 ...  
2  ridiculous      S1:2:6      Sensible      49  45_59 ...  
3  ridiculous      S1:2:6      Sensible      49  45_59 ...  
4  atrocious      A5:1      Evaluation:- Good/bad      49  45_59 ...
```

```
      n_speakers      list_speakers      rec_loc \  
0           3  S0588 S0589 S0590  ANON's home.  
1           3  S0588 S0589 S0590  ANON's home.  
2           3  S0588 S0589 S0590  ANON's home.  
3           3  S0588 S0589 S0590  ANON's home.  
4           3  S0588 S0589 S0590  ANON's home.
```

```
      relationships \  
0  Close family, partners, very close friends  
1  Close family, partners, very close friends  
2  Close family, partners, very close friends  
3  Close family, partners, very close friends  
4  Close family, partners, very close friends
```

```
      topics \  
0  Party, community market, maps, hospital appoin...  
1  Party, community market, maps, hospital appoin...  
2  Party, community market, maps, hospital appoin...  
3  Party, community market, maps, hospital appoin...  
4  Party, community market, maps, hospital appoin...
```

		activity	conv_type	conventions	\
0	Chatting whilst supper is cooking.	Discussing, Complaining	Revised		
1	Chatting whilst supper is cooking.	Discussing, Complaining	Revised		
2	Chatting whilst supper is cooking.	Discussing, Complaining	Revised		
3	Chatting whilst supper is cooking.	Discussing, Complaining	Revised		
4	Chatting whilst supper is cooking.	Discussing, Complaining	Revised		

	in_sample	transcriber
0	n	T10
1	n	T10
2	n	T10
3	n	T10
4	n	T10

[5 rows x 46 columns]

## 7.4 Write out

```
[41]: # speaker metadata
with open(f"out/metadata/speakers.csv", 'w') as f_out:
    speakers.to_csv(f_out)
```

```
[42]: # merged hits
with open(f"out/merged/{f_name_out}.csv", 'w') as f_out:
    df_merged.to_csv(f_out, index=False)
```