2-5. Antonyms– Collocates in Corpora (CC)

Antonym candidates – from Collocates in Corpora
Apply the co-occurrence hypothesis on corpora to generate antonym candidates. These candidates are tagged by linguists to generate antonyms and negation detection cue words. The general processes include:

- Discover collocate patterns from a corpus (the MEDLINE n-gram set)
- Generate antonym candidates from derived collocate patterns from the MEDLINE n-gram set
- Manually tagging
- Validate tags
- Update tags to annual release tag file

They are described in the following sections.

1. Discover collocate patterns from a corpus (the MEDLINE n-gram set)

1.1 Find collocate patterns from known aPairs and a corpus

- The input data we used for this task are:
  - aPairs: canonical aPairs from TtSet with the source of CC, randomly select aPairs as shown in Table 1.
  - corpus: the MEDLINE n-gram set, 2021

- Algorithm:
  Go through MEDLINE n-gram set and find all n-grams containing ant1 and ant2.
- To run the program:

To find collocates examples for given n-grams and an aPair.

```
Shell> cd ${ANTONYM}/bin
Shell> GetAntonyms ${YEAR}
60
```

- The output is at ${5.MEDLINE}/${YEAR}/output/Patterns.

  Table 1 shows results from collocate patterns from some selected aPairs using 2021 MEDLINE 3-gram set.

Ant-1	Ant-2	Collocate examples
normal	abnormal	• 11160|normal and abnormal • 1917|normal or abnormal • 463|abnormal and normal • 385|normal from abnormal • 243|normal versus abnormal • 159|normal to abnormal • 125|abnormal or normal • …
cool	warm	• 197|warm and cool • 130|cool and warm

		• 53|warm or cool
decrease	increase	• 5934|increase or decrease • 1990|increase and decrease • 940|decrease or increase • 691|decrease and increase • 205|decrease with increase • 167|increase, decrease, or
external	internal	• 15160|internal and external • 6836|external and internal • 1667|internal or external • 898|external or internal • 184|internal versus external • 124|internal to external • 122|internal, and external • 116|internal and/or external • 114|external to internal
large	small	• 12379|small and large • 8213|large and small • 1554|small or large • 1061|large or small • 883|small to large • 264|large to small • 185|small intestine, large • 169|large versus small • 153|small versus large
quick	slow	• 38|quick and slow • 37|slow and quick
sick	well	• 43|sick and well • 32|well as sick

Table 1. Collocate examples from the MEDLINE 3-grams, 2021

- Results:

We observed from table 1,

- Most of these aPairs fall into the collocate patterns of **[Ant-1 keyword Ant-2].** Keywords are in the middle of the 3-gram, including "and", "or", "versus", "to", etc.
- Some aPairs, such as calm|excited, buyer|seller, are not co-occurring in the MEDLINE 3-grams. The reasonable guess are:
  1) the MEDLINE n-gram set does not cover these aPairs. In such case, we suggest applying this collocates model with another corpus to find collocate patterns.
  2) These aPairs cannot be derived by collocate model. In such case, we suggest performing more research and focus on the semantics. These types of aPairs are categorized with source of [SN] (semantic in corpus).

- Other aPairs from experience:

There are other aPairs that are observed from our tagging processes, such as:

Ant1	Ant2	Collocate examples
extraordinary	ordinary	• Ordinary and extraordinary

The aPair of extraordinary|ordinary is from prefix dPairs. They cannot be derived from our prefixD model because they are not negation in the prefixD. However, they can be derived from the collocate model.

1.2 Find keywords in collocate patterns

Once we identify the pattern of [Ant1 keyword Ant2], the next step is to identify the most frequent keywords from a set of known aPairs. We develop computer programs to identify keywords using MEDLINE 3-grams and known canonical aPairs with source of CC from TtSet and antonyms from 2020 data.

- The input data we used for this task are:
  - aPairs: aPairs from the source of CC in the TtSet (antonyms.TtSet.data.${YEAR})
  - corpus: the MEDLINE 3-gram set, 2021 (3-gram.${YEAR}.30.core)
  - tagged file: antCandTtSet.data.tag.tagged.${YEAR}

- Algorithm:
  - Generate the canonical aPairs (include spVars) from TtSet and tagged candidate file.
  - Retrieve canonical aPairs with source of CC from above.
  - Split the canonical aPairs from CC into training (80%) and test (20%) set.
  - Get all keywords for the pattern of [ant1 keyword ant2] from training set.
    - Heuristic rules are implemented to reduce noise by excluding exceptions shown in table 2.

aPair	Examples of 3-grams that are not aPair patterns
from|to	• … range [from 1 to] 5 … • … [from baseline to] endpoint … • … [from childhood to] adolescence …
with|without	• … [with fever without] source … • … compared [with patients without] a history …
in|out	• … [in 11 out] of 17 … • … was found [in six out] of seven …
answer|question	• … [answer the question] … • … [answer that question] ...
external|internal	• … [external rotation internal] … • … [external iliac external] …
female|male	• … [female 2 male] 5 … • … [female four male] five …
length|width	• … [length x width] 5 … •
high|low	•
less|more	•

north|south	●
east|west	●

Table 2. Heuristic rules for aPair exceptions on retrieving keywords.

- To run the program:

To find keywords from canonical CC training set of TtSet in given n-grams.

```
Shell> cd ${ANTONYM}/bin
Shell> GetAntonyms ${YEAR}
61 (All Antonyms)|62 (TtSet)
```

- The output is at ${5.MEDLINE}/${YEAR}/output/Keywords/keyword.TtSet.cc.data.train
- Results:
  - Manually review the result and select the top 8 keywords by the highest word count for the patterns (Table 3).

ID	keywords	Accu. WC (All Antonyms)	Accu. WC (TtSet)	Select
1	And	389399	134618	Yes
2	Or	74669	34078	Yes
3	To	8981	4418	Yes
4	versus	7137	1489	Yes
5	than	3184	814	Yes
6	Vs	1781	401	Yes
7	from	1340	232	Yes
8	and/or	795	175	Yes
9	with	448	410	No

Table 3. Selected keywords for collocate pattern of [ant1 keyword ant2] from the MEDLINE 3-grams

2. Find antonym candidates from MEDLINE 3-grams

Antonym candidates are then generated by the CC pattern of [ant1 keyword ant2] from the MEDLINE 3-grams and derived keywords listed in table 3.

2.1 Generate antonym candidates from derived pattern with WC (option 65|66|67)

The algorithm applies the following criteria to filter out non-antonym candidates. These criteria are based on the CC pattern of [Ant1 keyword ant2] and the analysis in documents of 2-4-1.AntAnalysis-TT.

- n-grams matches certain patterns with terms. Such as:
  - 3-gram: [Ant1 vs Ant2], [Ant1 or Ant2], [Ant1 and Ant2],
  - 5-gram: [Ant1 as well as Ant2] (TBD)
- Ant1 and Ant2 must be in the Lexicon
- Ant1 and Ant2 must be single words
- Ant1 is not equal to Ant2
- Ant1 and Ant2 must have CUI

- Ant1 and Ant2 have common STI
- Ant1 and Ant2 are not synonyms
- Ant1 and Ant2 must has same POS: POS1=POS2
- retrieve citation, EUI, POS for Ant1 and Ant2 for antonym candidate list
- remove duplications (the citation form might be the same even if Ant1 and Ant2 are different, remove the aPair of citation)

This process is conducted for all selected (8) keywords from Table 3 (option 65|66) to generate ./PreCand/antCandPatMid.${KEYWORD}.data.wc. The format of the output file is the same as the format of the antonym candidate plus word count at the beginning, as shown in table 4.

Word Count	Ant1	EUI1	Ant2	EUI2	POS	Canon	Type	Negation	Domain	Source

Table 4. Field Format of Antonym Candidates from CC Pattern with 1 keyword

2.2 Generate antonym candidates by combining above results (option 67)
We further develop programs to combine results from the above steps with different keywords to calculate:
1) word count
2) keyword pattern count
for all antonym candidates generated from above steps with different keywords (antCandPatMid.all.data). The output file is then categorized into:

- antCandPatMid.all.data.tag: already tagged in our database
- antCandPatMid.all.data.tag.CC: already tagged with source of CC
- antCandPatMid.all.data.tbd: yet to be tagged, candidate file

To run the program,

```
Shell> cd  ${ANTONYM}/bin
Shell> GetAntonyms ${YEAR}
67
```

The output candidate file (antCandPatMid.all-3.data.tbd) has 11 fields, as shown in Table 5.

- 3-gram, from Pattern [Ant1 keyword Ant2]
  Where keywords include and, or, to, versus, than, vs, from, and/or

KC	**WC**	**Ant1**	**EUI1**	**Ant2**	**EUI2**	**POS**	**Canon**	**Type**	**Negation**	**Domain**
…	…	…	…	…	…	…	…	…	…	CC
8	77267	left	E0037123	rite	E0053604	adj				CC
8	73634	acute	E0007127	chronic	E0016869	adj				CC
8	43159	benign	E0012350	malignant	E0038686	adj				CC
…	…	…	…	…	…	…	…	…	…	CC
7	63545	primary	E0049990	secondary	E0054914	adj				CC
7	29338	anterior	E0234105	posterior	E0049132	noun				CC
7	11320	mild	E0040261	severe	E0055474	adj				CC

…	…	…	…	…	…	…	…	…	…	CC
6	49111	mild	E0040643	severe	E0055474	adj				CC
6	19679	less	E0037299	much	E0587115	adv				CC
6	15595	multiple	E0041326	single	E0056019	adj				CC
…	…	…	…	…	…	…	…	…	…	CC
5	37757	diastolic	E0022368	systolic	E0059731	adj				CC
5	14539	absolute	E0006593	relative	E0052609	adj				CC
…	…	…	…	…	…	…	…	…	…	CC

Table 5. Antonym Candidates from CC Pattern by combining 8 keywords

Three filters are observed to implement to increase precision for above model:
1). aPiars include "other|E0044444" (204)

- 6|5276|cell|E0015748|other|E0044444|noun|CANON_TBD|TYPE_TBD|NEG_TBD|DOMAIN_TBD|CC
- 5|5060|other|E0044444|patient|E0045987|noun|CANON_TBD|TYPE_TBD|NEG_TBD|DOMAIN_TBD|CC
- ..

2). Self-aPairs (4):

- 4|3655|cell|E0015748|cell|E0015748|noun|N|TYPE_TBD|O|DOMAIN_TBD|CC

*I know there are rare examples of 'self-antonyms' in English, like inflammable (able to be inflamed) and inflammable(not flammable), but because we don't separate senses in the Lexicon and these are both the same entry, they shouldn't quality as an aPair since we have no way to know which definition is intended at any given time.

3). Chose KC (keyword count is >= 3)

Also, only retrieve those with "|CANON_TBD|TYPE_TBD|" aPairs. Namely, to remove aPair are CC|PD|SD|LEX

The output file is: "antCandPatMid.cand.data.tbd". This file is used as candidate file.

The above candidate list is then sent to linguists to tag. The first field is the keyword pattern count. 8 means the associated candidates derived from the pattern with all 8 keywords. The second field is the accumulated word count from results of all 8 keywords. The rest of the fields are the same as the antonym candidate file from previous sections.

Future work

This is the result from our first CC model. There is more work for us to test, analyze, review, develop, including but not limiting to the following:

- To find out the precision and threshold for KC and WC for valid aPairs from the above result.
- To test the result on the test set of TtSet and find the performance.
- To derive other collocate patterns.

- To derive collocate patterns for 4-grams and 5-grams.
- To derive collocate patterns for 3-gram and in general

3. Tag Candidates (TBD)

Manual tagging is needed for the (new) antonym candidates generated from the above process. The tagged information of pre-existing candidates from previous years is saved and used as the baseline for future releases. Please refer to document 1.2.Antonym-Tags for details.

4. Validate and auto-fix Antonym Candidate Tags

Manual tags are verified by computer programs to:

- ensure all tags are valid
- automatically assign type to [NA] and domain to [DOMAIN_NONE] if Canon is [N]
- check for new domains.

Please refer to document 1-2.LexAntonym-Tag for details.

5. Update to Annual Release Antonym Tag file

- add tag result from source of CC to .${0.Antonym}/${YEAR}/input/antCand.data.tag.${YEAR}
- rerun the processes 1-4 until all candidates have valid tags (antCandPatMid.data.tbd = 0)

Please refer to document 1-2.LexAntonym-Tag for details.