

2-2. Antonyms – Source of SuffixD (SD)

Antonym candidates – from Suffix Derivation with negative tags

Negative tags in the Lexicon suffix derivations are used to generate antonym candidates. These candidates are tagged by linguists to generate antonyms and negation detection cue words. The general processes include:

- Generate antonym candidates from lexicon suffix derivations with negative tags
- Manually tagging
- Validate tags
- Update tags to annual release tag file

They are described in the following sections.

1. Generate Antonym Candidates

Antonym candidates are retrieved from the Lexicon derivation file (derivation.data.{\$YEAR}). These antonym candidates are retrieved from the following suffix rules with the POS of adj:

- \$|noun|less\$|adj
- \$|verb|less\$|adj

1.1 Algorithm

- derivation.data includes 9 fields as shown in table 1:

Derivation1	POS1	EUI1	Derivation2	POS2	EUI2	Negation	type	prefix
-------------	------	------	-------------	------	------	----------	------	--------

Table 1. Format in the Lexicon derivation file.

- Retrieve candidate as follows:
 - Get negative suffix derivation
 - Prefix: type=S
 - Negative: negation=N
 - Derivation1 is a legal single word
 - Derivation2 is a legal single word (citation, not spVar)
 - Derivation2 is a citation (not a base of spelling variants)
 - Find derivation of Derivation1 and form antonym candidate
 - Use EUI1 as key to find derivation (derivation3) of Derivation1 that:
 - Has same POS (POS3) as POS2 (adj)
 - Derivation3 is a citation (not spVar)
 - Derivation3 != Derivation2, not Derivation2 (itself)
 - Format:

Derivation3	EUI3	Derivation2	POS2	EUI2	Canon	type	Negation	Domain	Source
-------------	------	-------------	------	------	-------	------	----------	--------	--------

▪

○ Example:

- Get negative suffix derivation

```
...
count | noun | E0019376 | countless | adj | E0019416 | N | S | None
count | verb | E0019375 | countless | adj | E0019416 | N | S | None
...
```

- Find derivation of Derivation1 and form antonym candidate:

Two EUIs:

```
E0019376 -> countless | adj | E0019416
```

- ⇒ Only has 1 derivation3, however, it is the same as derivation2
- ⇒ No antonym candidate is retrieved for E0019376

```
E0019375 -> countable | adj | E0019378 (generate candidate)
```

```
-> counter | noun | E0019381 (POS3 != POS2, not a candidate)
```

```
-> countless | adj | E0019416 (Derivation3 = Derivation2, not a candidate)
```

- ⇒ One antonym candidate is retrieved for E0019375

```
countable | E0019378 | countless | E0019416 | adj
```

1.2 Program - Derivation.GetAntCandFromSuffixD.java

- Input:
 - Link ./2.SuffixD/\${YEAR}/input/derivation.data
 - Link ./1.Lexicon/\${YEAR}/input/infVars.data
 - Update ./0.antonym/input/antCand.data.\${YEAR}.tag (the latest annual release antonym candidates tagged file)
 - Update ./0.antonym/input/domain.data

```
Shell> cd ${ANTONYM}/bin
```

```
Shell> GetAntonyms ${YEAR}
```

```
20
```

- output files:
 - ./output/antCandSuffixD.data

- ./output/antCandSuffixD.data.tag
- ./output/antCandSuffixD.data.tbd (candidate file)
- ⇒ If this file is not 0, use this file as new candidates and send it to linguists to tag

1.3 Output – Candidate Format and Examples

The output candidate file has 10 fields, as shown in Table 2. The blank cells are automatically tagged with [XXX_TBD], where XXX are CANON, TYPE, NEG and DOMAIN.

Ant1	EUI1	Ant2	EUI2	POS	Canon	Type	Negation	Domain	Source
useful	E0063740	useless	E0063743	adj					SD
countable	E0019378	countless	E0019416	adj					SD
colorable	E0523388	colorless	E0017911	adj					SD
colored	E0071157	colorless	E0017911	adj					SD
colorful	E0017909	colorless	E0017911	adj					SD
...					SD

Table 2. Examples of antonym candidates from Lexicon suffix derivation with negative tags

2. Tag Candidates

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.

3. Validate and Auto-fix Antonym Candidates

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.

4. Update to Overall Antonym Tag file

- add tag result from source of SD to .\${0.Antonym}/\${YEAR}/input/antCand.data.tag.\${YEAR}
- rerun the processes 1-3 until all candidates have valid tags (antCandSuffixD.data.tbd = 0)

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