Lab notebook

KBC

4/5/2021

2021-04-05 report

Utility functions

Opposites already manually paired

Bill can explain where these came from better than I can.

Comparison to Chris et al.'s stuff

OK, we know the intersection/complements, so let's see what Chris's opposites look like, exactly.

RO exploration

I extracted the terms from the current version of the RO, then got the absolute frequencies of their words:

/Users/kevincohen/Dropbox/Scripts-new/lexicalFrequency.pl experimental-outputs/ro.terms.txt

. . . which reveals imbalances in the use of words that are clear opposites, e.g.:

- during 11 before 4 after 2
- to 49 from 16 towards 1 *how about away?*
- directly 18 indirectly 8
- indirect 2 direct 2

. . . which suggests that I could be adding a bunch of opposite terms to the ontology.

So, I ran the current version of my antonym-finding script:

code/findAntonyms.pl resources/ontologies/ro.2021-03-08.obo.txt |wc -l

. . . which finds 43 pairs of opposites.

Next step: find overlap with Mike's resources/ontologies/predicates.txt.

For next weeks. . .

0. With Bill, reorganize repository and merge with TRANSLATOR's
1. Test cases in human-readable form
2. Handle substitutions (e.g. hypercalcemia vs. hypocalcemia)
3. Suffixes (e.g. hydrophilic vs. hydrophobic)
4. Word-internal, possibly (might produce a lot of false positives later when we do generation)

2021-04-15 report

Analysis of PATO opposites WRT the excluded middle

Motivation: this picks out a specific type of opposition: what Pustejovsky calls *polar*, meaning that there's a scale and the things at the two poles of the scale are opposites of each other.

How I assigned the excluded.middle value

0. If the middle is reasonably clearly excluded, I assigned the value *yes*. Examples: *acute/chronic, aerobic/anaerobic*. If that is not the case, then I put the "middle" value in the excluded.middle field. Example: for *phosphorylated/dephosphorylated*, I put the value *unphosphorylated* in the excluded.middle column.
1. Almost every pair that fits the pattern *increased/decreased x* has a mid-point or neutral point *normal x*.
2. Most pairs of the form *x/unx* (e.g. *responsive/unresponive*) exclude the middle. Exception: *damaged/undamaged/repaired*.
3. Most pairs of the form *x/dex* (e.g. *phosphorylated/dephosphorylated*) have a neutral point *unx* (e.g. *unphosphorylated*).
4. Most pairs of the form *hypox/hyperx* where both members of the pair are single words have a single-word neutral point *normox*. For example, for the pair *hypotrophic/hypertrophic*, there is a neutral point *normotrophic*. If I had any question about the legitimacy of these, I checked Google Scholar to ensure that the *normox* word is used.
5. If I did not find such a word via Google Scholar, then I searched for the phrase *neither term01 nor term02*. If I found it, then I put the phrase in the field. (TODO: now that I think about it, if I didn't find the phrase, I didn't try again with the order *term02 term01*. Need to do that.)

```
# "current" is the ones that Bill found currently in use
in.both <- read.table("/Users/kevincohen/Dropbox/N-Z/translator-concept-oppositeness/experimental-outpu
in.mine.only  <- read.table("/Users/kevincohen/Dropbox/N-Z/translator-concept-oppositeness/experimental
in.current.only <- read.table("/Users/kevincohen/Dropbox/N-Z/translator-concept-oppositeness/experimenta

in.both$excluded.middle <- as.character(in.both$excluded.middle)
in.mine.only$excluded.middle <- as.character(in.mine.only$excluded.middle)
in.current.only$excluded.middle <- as.character(in.current.only$excluded.middle)
# in.both <- as_tibble(in.both)
# in.both <- mutate(in.both, found.by = "BOTH")
# in.mine.only <- as_tibble(in.mine.only)
# in.mine.only <- mutate(in.mine.only, found.by = "ME")
# in.current.only <- as_tibble(in.current.only)
# in.current.only <- mutate(in.current.only, found.by = "CURRENT")

# in.combined <- as_tibble(in.both, in.mine.only, in.current.only)

#ggplot(data = in.combined, mapping = aes(x = found.by, )) +
#  geom_bar(stat ="identity")

in.counts <- c(nrow(in.both), nrow(in.mine.only), nrow(in.current.only))
#in.counts <- as_tibble(in.counts, counts = in.counts)
barplot(in.counts, names.arg = c("BOTH", "ME ONLY", "CURRENT ONLY"))
```

```
# I CAN'T GET THIS TO WORK...
#excluded.middle.counts <- c()
##count.both <- nrow(select(in.both, in.both$excluded.middle == "yes"))
##select(in.both, in.both$excluded.middle == "yes")
##count.both <- nrow(in.both$term01[in.both$term01 == "yes"])
##count.both <- in.both[which(in.both$excluded.middle == 'yes')]

#count.mine <- nrow(in.mine.only$term01[in.mine.only$term01 == "yes"])
#count.current <- nrow(in.current.only$term01[in.current.only$term01 == "yes"])
#excluded.middle.counts <- c(count.both, count.mine, count.current)

#barplot(excluded.middle.counts, names.arg = c("BOTH", "ME ONLY", "CURRENT ONLY"))
```

For next weeks...

1. Test cases in human-readable form
2. Handle substitutions (e.g. hypercalcemia vs. hypocalcemia)
3. Failing test case: *protein folding* versus *protein unfolding*
4. Failing test case: *name: oil gland decreased thickness* versus *oil gland increased thickness*
5. Suffixes (e.g. hydrophilic vs. hydrophobic)
6. Consistent naming scheme for experimental-results directory
7. Word-internal, possibly (might produce a lot of false positives later when we do generation)
8. *See email exchange with Bill*

2021-04-21 report

Now handling:

1. Now handling morphological substitutions, as opposed to additions. That means that where we used to get only pairs like abnormal, where the contrast is between presence of ab and absence of ab, we now also get hypercalcemia/hypocalcemia, where the contrast is not presence/absence, but rather between two things that are... Shit, I'm tired of trying to squeeze this into non-technical language. We used to only get prefix + free morpheme; now we are getting prefix + bound morpheme. Honi soit qui mal y pense.
2. Now handling suffixes. So, we now get pairs like hydrophilic/hydrophobic (thanks to Leslie Rapp for that one). Embarrassingly, I am not getting leukemia/leukopenia–bug being hunted.
3. Test cases in human-readable form
4. Failing test case: *protein folding* versus *p274rotein unfolding*
5. Failing test case: *name: oil gland decreased thickness* versus *oil gland increased thickness*
6. Consistent naming scheme for experimental-results directory
7. *See email exchange with Bill*
8. Word-internal, possibly (might produce a lot of false positives later when we do generation)

2021-05-05 report

1. Took a week of vacation
2. Took a sick day
3. Generated the outputs for all of the CRAFT ontologies, plus HPO, MPO, and PATO

2021-05-26

```
# only need to do once
#install.packages("entropy")
#library(entropy)

mi.calcs <- read.table("/Users/kevincohen/Dropbox/N-Z/translator-concept-oppositeness/experimental-outpu

get.rid.of.commas <- function(input.vector) {
  output.vector <- gsub(",", "", input.vector)
  return(output.vector)
}

# preprocessing--some things need to be integers, others factors
mi.calcs$opposite <- factor(mi.calcs$opposite)
mi.calcs$x.count <- as.integer(get.rid.of.commas(mi.calcs$x.count))
mi.calcs$y.count <- as.integer(get.rid.of.commas(mi.calcs$y.count))
mi.calcs$y.minus.x.count <- as.integer(get.rid.of.commas(mi.calcs$y.minus.x.count))
mi.calcs$x.minus.y.count <- as.integer(get.rid.of.commas(mi.calcs$x.minus.y.count))
mi.calcs$xy.count <- as.integer(get.rid.of.commas(mi.calcs$xy.count))
as_tibble(mi.calcs)


#mi.calcs <- mi.calcs %>% mutate(p.x = (x.count / (x.count + y.minus.x.count))) %>% mutate(p.y = (y.cou
#mutate(mi.or.something = p.xy / (p.x * p.y))

#mi.calcs <- mi.calcs %>% mutate(p.x = x.count / (x.not.y.count + y.not.x.count + x.and.y.count))
#mi.calcs <- mi.calcs %>% mutate(p.x = x.count)
# calculate p(x)
mi.calcs <- mi.calcs %>% mutate(p.x = x.count / (x.minus.y.count + y.minus.x.count + xy.count))
# calculate p(y)
mi.calcs <- mi.calcs %>% mutate(p.y = y.count / (x.minus.y.count + y.minus.x.count + xy.count))
# calculate p(x,y)
mi.calcs <- mi.calcs %>% mutate(p.xy = xy.count / (x.minus.y.count + y.minus.x.count + xy.count))
mi.calcs <- mi.calcs %>% mutate(mi.or.something = p.xy / (p.x * p.y))

#mi.calcs$mi.or.something <- log(mi.calcs$mi.or.something)

# Here I plot the values on a scale from 0.0 to 1.0
ggplot(data = mi.calcs, mapping = aes(x = opposite, y = mi.or.something)) +
  geom_boxplot() +
  ylim(0, 1.0) +
  #ylab("MI or something") +
  labs(x = "Non-opposites versus opposites", y = "MI or something", title = "Like MI but not log")
  #names(c("Not opposites", "Opposites"))

shapiro.test(mi.calcs$mi.or.something)

# Here I plot their logs
mi.calcs$mi.or.something <- log(mi.calcs$mi.or.something)
ggplot(data = mi.calcs, mapping = aes(x = opposite, y = mi.or.something)) +
  geom_boxplot() +
  #ylim(0, 1.0) +
```

```
  #ylab("MI or something") +
  labs(x = "Non-opposites versus opposites", y = "MI or something", title = "MI (maybe)")
  #names(c("Not opposites", "Opposites"))

head(mi.calcs)
#mi.calcs$mi.or.something %>% gather(opposite)

#mi.calcs <- mi.calcs %>%  select(opposite, mi.or.something)
#head(mi.calcs)

#wilcox.test(select(mi.calcs$opposite == "0"), select(mi.calcs$opposite == "1"))
```

Next step

Multi-word phrases, moving towards opposite sides of normal

HPO/MPO/MONDO terms with increase and decrease; then adjectival ones (especially hyper- and hypo-) (and generate more of those? Easy enough to do)

- Variability across those? Like, increased versus "increase in", "elevation of", etc.? Again, it's generation...
- Additional affixes: over- and under-

2021-06-02

1. Observation: synonymy is impoverished in these ontologies. For example, *increased hemoglobin* is probably equivalent to *elevated hemoglobin*, but only the first is in the ontology.
2. So, the data would benefit from Dr. Funking.
3. Logical opposites don't necessarily occur in these ontologies, and there might be good reasons for that. For example, *hypoxemia* is a clearly clinically relevant concept, but *hyperoxemia* might *not* be. This contrasts with *decreased hemoglobin affinity for oxygen* and *increased hemoglobin affinity for oxygen*, both of which *are* entirely clinically relevant.

Here are some numbers that support (1) and (3):

HPO contains:

- 459 non-obsolete terms with 'increased'
- 370 non-obsolete terms with 'decreased'
- 168 paired non-obsolete increased/decreased terms

This suggests that although perhaps there should *not* be more pairs of opposites, there certainly *could* be.

- 453 non-obsolete terms with 'hyper'
- 671 non-obsolete terms with 'hypo'
- 108 paired non-obsolete hypo/hyper terms

Again, this suggests that although perhaps there should *not* be more pairs of opposites, there certainly *could* be.

- 165 with 'reduced'

- 153 with 'elevated'
- 4 with 'depressed'
- 33 with 'high'
- 61 with 'low'

The 153:33 ratio of *elevated* to *high* and 165:4:61 of *reduced/depressed/low* suggests that for recognition in text, Funkification would increase recall.

So, for any given ontology, here's what I did:

1. Grep out the terms with *increase, decrease, hyper,* or *hypo.*
2. Find the subset of those (increase/decrease and hyper/hypo) that do occur in addition to their logical opposite. (scripts: *increasedDecreasedOpposites.pl* and *hyperHypoOpposites.pl*)
3. For that subset, generate synonyms for both members of the pair. (script: *generatePairs.pl*)

Now let's go to a terminal...

2021-06-09

I've updated the format of the files that contain opposites and synonyms. Now it encodes whether or not they're opposites, whether or not they're synonyms (you can be one, or the other, or neither, but not both); whether they're in the original form (i.e. exact match to the term in the ontology), or derived; and their source.

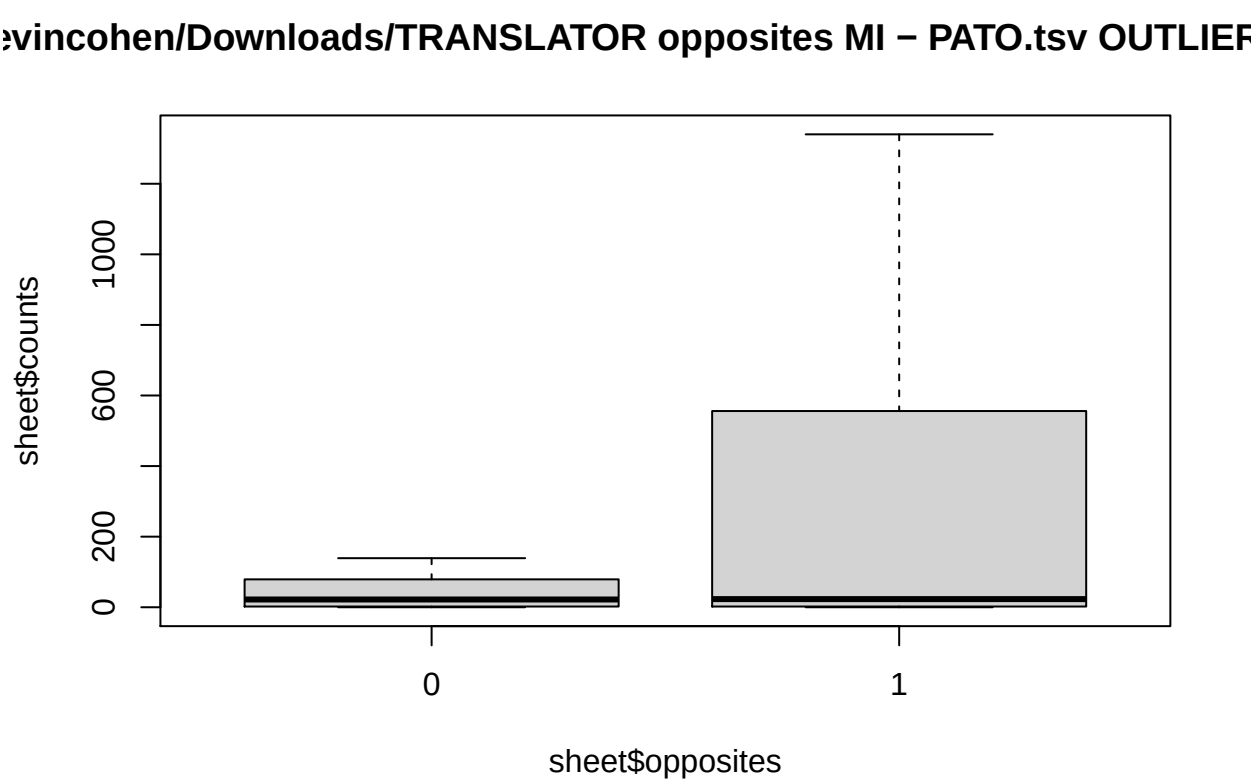
So, this code takes those and automates the searches that last week I was doing manually.

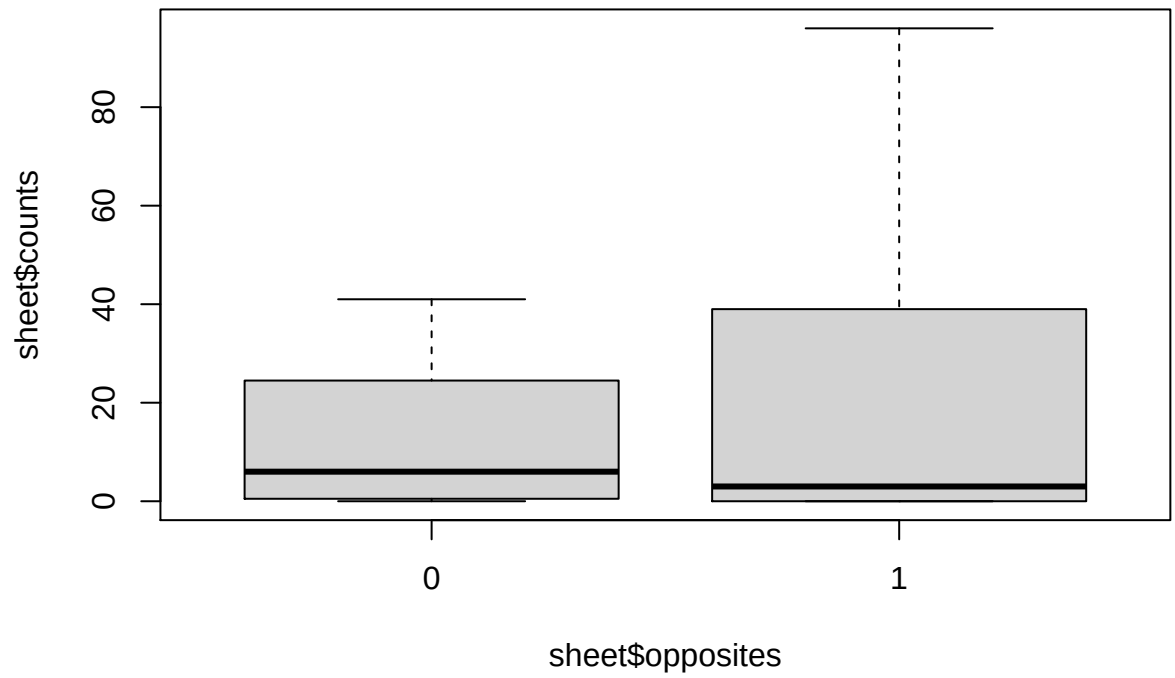
```
library("easyPubMed")
```

```
## Warning: package 'easyPubMed' was built under R version 4.0.2
```

```
files <- c("/Users/kevincohen/Downloads/TRANSLATOR opposites MI - PATO.tsv",
           "/Users/kevincohen/Downloads/TRANSLATOR opposites MI - HPO.tsv")

#sheet <- read.table("/Users/kevincohen/Downloads/TRANSLATOR opposites MI - PATO.tsv",
#                    header = TRUE, sep = "\t")

counts <- as.integer(c())
sheet <- data.frame()

for (file_number in 1:length(files)) {
  if (DEBUG) { print(paste("File:", files[file_number])) }
  sheet <- NULL
  counts <- NULL
  # stupid name for the variable, derived from the fact that
  # I made the files from different sheets of the same spreadsheet
  sheet <- data.frame()
  sheet <- read.table(files[file_number], header = TRUE, sep = "\t")
  sheet$opposites <- as.factor(sheet$opposites)
```

```
  # FOR DEV ONLY
  #sheet <- sheet[1:5, ]

  for (i in 1:nrow(sheet)) {
  #for (i in 1:5) {

    if (DEBUG) { print(paste("Row number:", i))}

    my_query <- paste('"', sheet[i, "term.01"], '"', " ",  '"', sheet[i, "term.02"], '"', sep = "")
    #my_query <- paste('"', sheet[i, "term.01"], '"', " ",  '"', sheet[i, "term.02"], sep = "")
    if (DEBUG) { print(my_query) }
    my_entrez_id <- get_pubmed_ids(my_query)
    #print(my_entrez_id$Count)
    counts <- c(counts, as.integer(my_entrez_id$Count))
  } # loop through pairs of terms

  column.names <- colnames(sheet)
  if (DEBUG) { print("Add counts to data.frame")
            print(paste("Rows in data.frame:", nrow(sheet), "Elements in counts:", length(counts)))}
  sheet <- cbind(sheet,counts)
  column.names <- c(column.names, "counts")
  if (DEBUG) { print("Reset column names") }
  colnames(sheet) <- column.names

  if (DEBUG) { print("Finally, generate the graph") }
  boxplot(sheet$counts ~ sheet$opposites,
          #main = files[file_number],
          main = paste(files[file_number], "OUTLIERS REMOVED"),
          outline = FALSE) # outline = FALSE removes outliers, of which there are quite a few

  wilcox.test(sheet$counts ~ sheet$opposites)

} # close for-loop through list of files
```


evincohen/Downloads/TRANSLATOR opposites MI − PATO.tsv OUTLIEF
sheet$counts
0
200
600
1000
0
1
sheet$opposites




```
#print(counts)

#boxplot(counts)
```

For reproducibility

```
## R version 4.0.1 (2020-06-06)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS  10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] easyPubMed_2.13     ggVennDiagram_0.5.0 forcats_0.5.1
##  [4] stringr_1.4.0       dplyr_1.0.4         purrr_0.3.4
##  [7] readr_1.4.0         tidyr_1.1.1         tibble_3.0.1
## [10] ggplot2_3.3.2       tidyverse_1.3.0
```

```
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.4.6         lubridate_1.7.9      class_7.3-17
##  [4] assertthat_0.2.1     digest_0.6.25        R6_2.4.1
##  [7] cellranger_1.1.0     futile.options_1.0.1 backports_1.1.8
## [10] reprex_1.0.0         evaluate_0.14        e1071_1.7-3
## [13] highr_0.8            httr_1.4.1           pillar_1.4.4
## [16] rlang_0.4.10         readxl_1.3.1         VennDiagram_1.6.20
## [19] rstudioapi_0.11      rmarkdown_2.8        munsell_0.5.0
## [22] broom_0.7.4          compiler_4.0.1       modelr_0.1.8
## [25] xfun_0.23            pkgconfig_2.0.3      htmltools_0.5.1.1
## [28] tidyselect_1.1.0     fansi_0.4.1          crayon_1.3.4
## [31] dbplyr_2.1.0         withr_2.4.1          sf_0.9-8
## [34] grid_4.0.1           jsonlite_1.7.0       gtable_0.3.0
## [37] lifecycle_0.2.0      DBI_1.1.0            magrittr_1.5
## [40] formatR_1.8          units_0.7-1          scales_1.1.1
## [43] KernSmooth_2.23-17   cli_2.0.2            stringi_1.4.6
## [46] fs_1.4.1             xml2_1.3.2           futile.logger_1.4.3
## [49] ellipsis_0.3.1       generics_0.0.2       vctrs_0.3.6
## [52] lambda.r_1.2.4       tools_4.0.1          glue_1.4.1
## [55] hms_1.0.0            yaml_2.2.1           colorspace_1.4-1
## [58] classInt_0.4-3       rvest_0.3.6          knitr_1.33
## [61] haven_2.3.1
```