Parents Voices Matter: A Mixed-Method Study on the Dyslexia Diagnosis Process

Select data

```r
library(stringr)

Q49 <- dp_tbl_text %>%
    select(id, state, Q49_text) %>%
    filter(str_detect(Q49_text, "")) %>%
    mutate(participant = row_number())

Q51 <- dp_tbl_text %>%
    select(id, state, Q51_text) %>%
    filter(str_detect(Q51_text, ""))

Q53 <- dp_tbl_text %>%
    select(id, state, Q53_text) %>%
    filter(str_detect(Q53_text, ""))

Q55 <- dp_tbl_text %>%
    select(id, state, Q55_text) %>%
    filter(str_detect(Q55_text, ""))

Q57 <- dp_tbl_text %>%
    select(id, state, Q57_text) %>%
    filter(str_detect(Q57_text, ""))

Q78 <- dp_tbl_text %>%
    select(id, state, Q78_text) %>%
    filter(str_detect(Q78_text, ""))
```

```r
library(tidyr)
library(stringr)

united_tbl <- dp_tbl_text %>%
    unite(united_texts, Q49_text, Q51_text, Q53_text, Q55_text, Q57_text, Q78_text, sep = "", remove = T
```

```r
united_tbl_clean <- united_tbl %>%
    select(id, state, united_texts) %>%
    filter(str_detect(united_texts, "")) %>%
    mutate(participant = row_number())

write.csv(united_tbl_clean, "../data/united_tbl_clean.csv")
```

```
library(stringr)

united_tbl_clean_tk <- united_tbl_clean

united_tbl_clean_tk$united_texts <- united_tbl_clean$united_texts  %>%
    str_extract_all(pattern = "[[:alpha:]']+-?[[:alpha:]']+|[:alpha:]{1}") %>%
    str_to_lower()
```

```
library(quanteda)
library(textminingR)
process.data <- united_tbl_clean_tk %>%
    preprocess_texts(text_field = "united_texts", remove_hyphens = FALSE) %>%
    tokens_wordstem() %>%
    tokens_select(min_nchar=3L,
                  verbose = TRUE)
```

```
library(quanteda)
dfm_output <- dfm(process.data)
dfm_output@docvars$document <- dfm_output@docvars$docname_
```

```
library(tidytext)
dfm_td_init <- tidy(dfm_output)
dfm_td <- dfm_td_init %>% left_join(dfm_output@docvars, "document")
dfm_td
```

```
## # A tibble: 8,592 x 9
##    document term   count docname_ docid_   segid_ id            state participant
##    <chr>    <chr>  <dbl> <chr>    <fct>     <int> <chr>         <chr>       <int>
##  1 text1    forc       1 text1    text1         1 R_3qqwNjMuDX~ SC              1
##  2 text29   forc       1 text29   text29        1 R_2D7zezWo9w~ WA             29
##  3 text59   forc       1 text59   text59        1 R_1ml4boLfUC~ FL             59
##  4 text82   forc       1 text82   text82        1 R_3EXbZr8T1k~ IN             82
##  5 text284  forc       1 text284  text284       1 R_3CWnyUiRIU~ SC            284
##  6 text315  forc       1 text315  text315       1 R_1LOwDftKTv~ TX            315
##  7 text1    school     1 text1    text1         1 R_3qqwNjMuDX~ SC              1
##  8 text4    school     2 text4    text4         1 R_1mLBLyBBGm~ TX              4
##  9 text7    school     1 text7    text7         1 R_1jersRUTbm~ TX              7
## 10 text10   school     6 text10   text10        1 R_3oF8rNi5Et~ VA             10
## # ... with 8,582 more rows
```

```
dfm_td$word <- dfm_td$term
```

Term frequency-inverse document frequency (tf-idf) of participants' responses

```
dfm_td_tf_idf <- dfm_td %>%
  bind_tf_idf(word, participant, count)
dfm_td_tf_idf
```

```
## # A tibble: 8,592 x 13
##    document term   count docname_ docid_  segid_ id      state participant word
##    <chr>    <chr>  <dbl> <chr>    <fct>    <int> <chr>   <chr>       <int> <chr>
##  1 text1    forc       1 text1    text1        1 R_3qqw~ SC              1 forc
##  2 text29   forc       1 text29   text29       1 R_2D7z~ WA             29 forc
##  3 text59   forc       1 text59   text59       1 R_1ml4~ FL             59 forc
##  4 text82   forc       1 text82   text82       1 R_3EXb~ IN             82 forc
##  5 text284  forc       1 text284  text284      1 R_3CWn~ SC            284 forc
##  6 text315  forc       1 text315  text315      1 R_1LOw~ TX            315 forc
##  7 text1    school     1 text1    text1        1 R_3qqw~ SC              1 scho~
##  8 text4    school     2 text4    text4        1 R_1mLB~ TX              4 scho~
##  9 text7    school     1 text7    text7        1 R_1jer~ TX              7 scho~
## 10 text10   school     6 text10   text10       1 R_3oF8~ VA             10 scho~
## # ... with 8,582 more rows, and 3 more variables: tf <dbl>, idf <dbl>,
## #   tf_idf <dbl>
```

Positive and negative words

```
word_counts_sentiments <- dfm_td_tf_idf %>%
    inner_join(get_sentiments("bing"))

word_counts_sentiments_selected <- word_counts_sentiments %>%
    dplyr::select(document, participant, id, state, word, count, sentiment, tf, idf, tf_idf)
word_counts_sentiments_selected
```

```
## # A tibble: 827 x 10
##    document participant id      state word  count sentiment      tf   idf tf_idf
##    <chr>          <int> <chr>   <chr> <chr> <dbl> <chr>       <dbl> <dbl>  <dbl>
##  1 text3              3 R_1gGf~ TX    susp~     1 negative  0.167    3.16 0.526
##  2 text62            62 R_2R9x~ MN    susp~     2 negative  0.0769   3.16 0.243
##  3 text68            68 R_ysjz~ TX    susp~     1 negative  0.5      3.16 1.58
##  4 text95            95 R_1r0v~ TX    susp~     1 negative  0.0556   3.16 0.175
##  5 text98            98 R_bsgc~ AZ    susp~     1 negative  0.0217   3.16 0.0686
##  6 text164          164 R_puFr~ TX    susp~     1 negative  0.0204   3.16 0.0644
##  7 text169          169 R_1jev~ TX    susp~     1 negative  0.1      3.16 0.316
##  8 text204          204 R_28HV~ VA    susp~     1 negative  0.00980  3.16 0.0310
##  9 text240          240 R_2tDp~ TX    susp~     1 negative  0.0556   3.16 0.175
## 10 text245          245 R_1ilp~ MA    susp~     1 negative  0.0179   3.16 0.0564
## # ... with 817 more rows
```

```
word_tf_idf <- word_counts_sentiments_selected %>%
    count(word, sentiment, tf_idf, sort = TRUE) %>%
    ungroup()

word_tf_idf
```

```
## # A tibble: 760 x 4
##    word    sentiment tf_idf     n
##    <chr>   <chr>      <dbl> <int>
##  1 work    positive  0.0651     6
##  2 better  positive  0.0490     3
##  3 right   positive  0.0745     3
##  4 succeed positive  0.150      3
##  5 success positive  0.150      3
##  6 support positive  0.0389     3
##  7 support positive  0.112      3
##  8 well    positive  0.0609     3
##  9 work    positive  0.0363     3
## 10 work    positive  0.0459     3
## # ... with 750 more rows
```

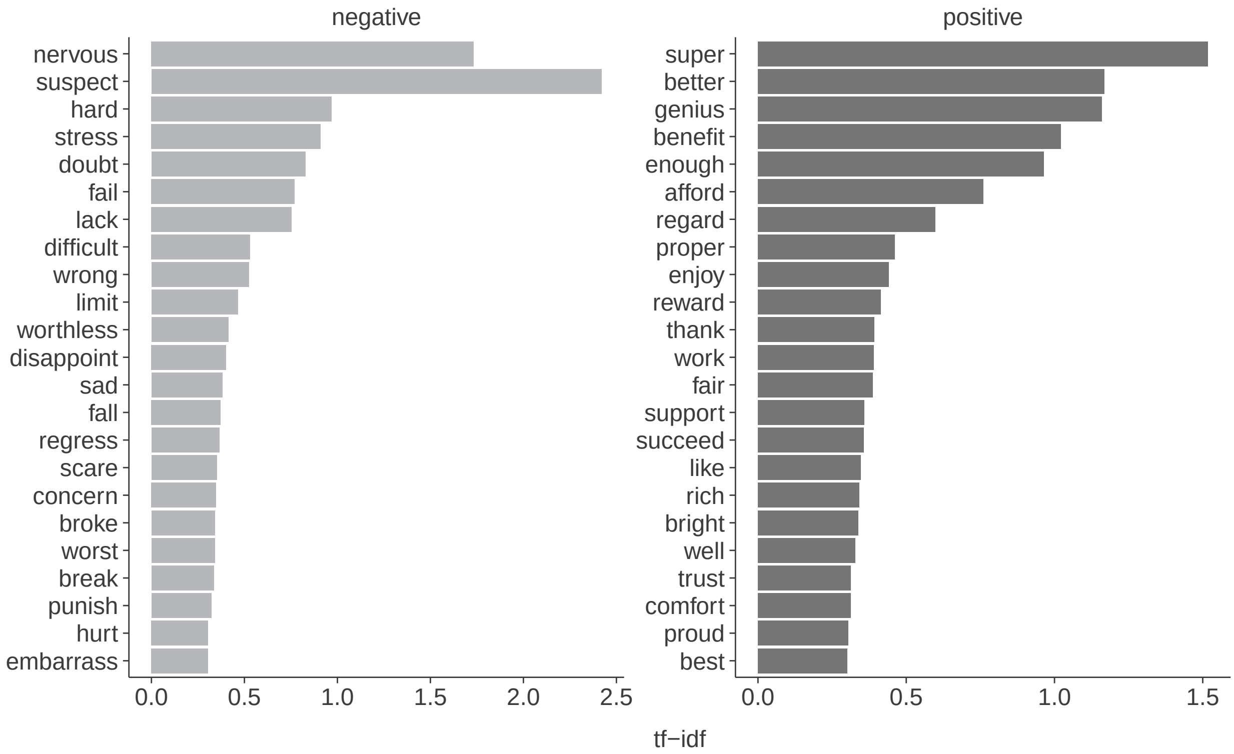
```
library(ggplot2)
sliced_sentiment <- word_tf_idf %>%
    filter(sentiment == "negative" & tf_idf >= 0.3 | sentiment == "positive" & tf_idf >= 0.3) %>%
    arrange(desc(tf_idf), word) %>%
    group_by(sentiment) %>%
    ungroup()

sliced_sentiment$word <- as.factor(sliced_sentiment$word)
sliced_sentiment$sentiment <- as.factor(sliced_sentiment$sentiment)
sliced_sentiment
```

```
## # A tibble: 56 x 4
##    word    sentiment tf_idf     n
##    <fct>   <fct>      <dbl> <int>
##  1 suspect negative   1.58      1
##  2 super   positive   1.17      1
##  3 genius  positive   1.16      1
##  4 nervous negative   1.10      1
##  5 benefit positive   1.02      1
##  6 hard    negative   0.969     1
##  7 doubt   negative   0.828     1
##  8 fail    negative   0.770     1
##  9 nervous negative   0.630     1
## 10 regard  positive   0.598     1
## # ... with 46 more rows
```

```
library(forcats)
library(tidytext)

sliced_sentiment_group_by <- sliced_sentiment %>%
    ungroup()%>%
    group_by(word,sentiment,n)%>%
    summarise(tf_idf=sum(tf_idf), .groups = 'drop') %>%
    ungroup()

sliced_sentiment_group_by
```

```
## # A tibble: 47 x 4
##    word      sentiment     n tf_idf
##    <fct>     <fct>     <int>  <dbl>
##  1 afford    positive      1  0.760
##  2 benefit   positive      1  1.02
##  3 best      positive      1  0.301
##  4 better    positive      1  1.17
##  5 break     negative      1  0.336
##  6 bright    positive      1  0.338
##  7 broke     negative      1  0.341
##  8 comfort   positive      1  0.313
##  9 concern   negative      1  0.347
## 10 difficult negative      1  0.531
## # ... with 37 more rows
```

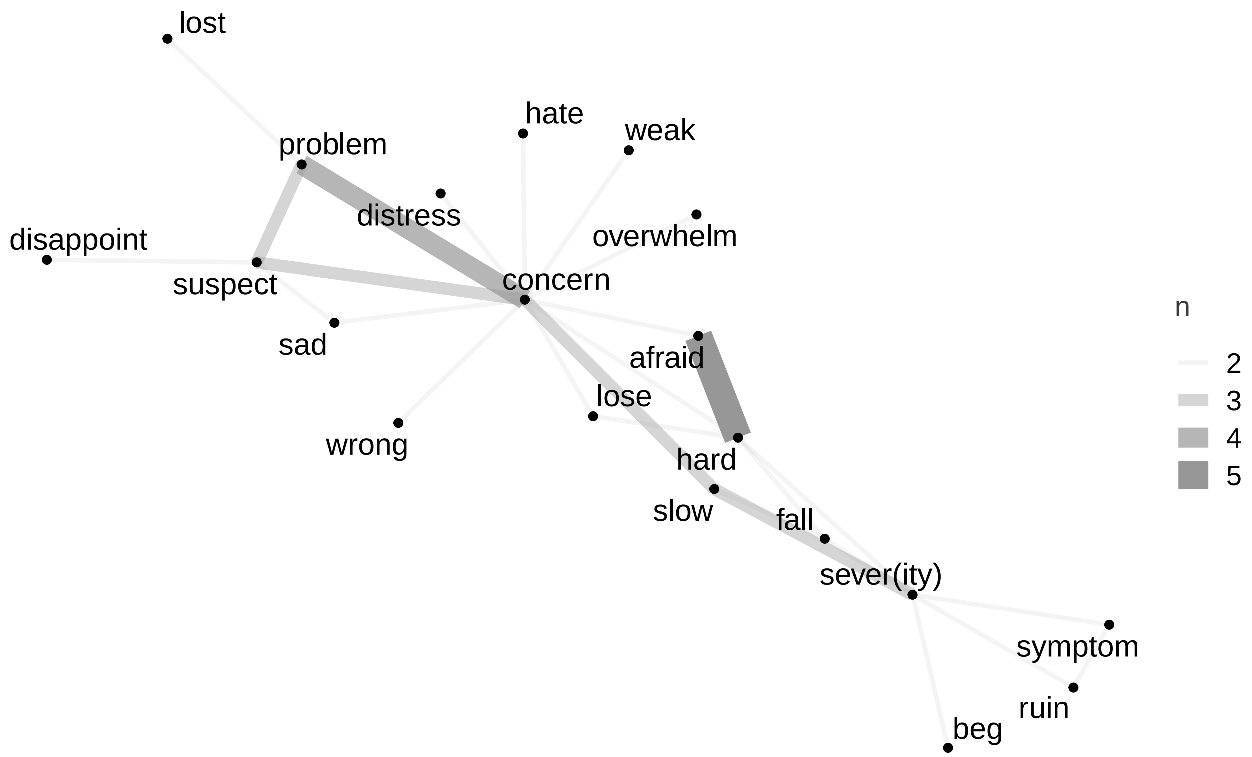
```
plot_sentiment <-  sliced_sentiment_group_by %>%
    mutate(word=as.character(word), word=reorder_within(word,tf_idf, sentiment)) %>%
    ggplot(aes(x = tf_idf, y = word, fill = sentiment)) +
    scale_fill_manual(values=c("#B5B7BB", "#757575")) +
    geom_col(show.legend = FALSE)+
    facet_wrap(~ sentiment, scales = "free") +
    scale_y_reordered()+
    labs(x = "tf-idf",
         y = NULL,
         size = 11) +
    theme_minimal(base_size = 11) +
    theme(
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line(color = "#3B3B3B", size = 0.3),
        axis.ticks = element_line(color = "#3B3B3B", size = 0.3),
        strip.text.x = element_text(size = 11, color = "#3B3B3B"),
        axis.text.x = element_text(size = 11, color = "#3B3B3B"),
        axis.text.y = element_text(size = 11, color = "#3B3B3B"),
        axis.title = element_text(size = 11, color = "#3B3B3B"),
        axis.title.x = element_text(margin = margin(t = 9)),
        axis.title.y = element_text(margin = margin(r = 9)))

plot_sentiment
```



```r
library(widyr)
word_neg <- dfm_td %>%
    inner_join(get_sentiments("bing")) %>%
    filter(sentiment == "negative") %>%
    pairwise_count(word, participant, sort = TRUE, upper = FALSE) %>%
    tibble() %>%
    ungroup()

word_neg$item1<- dplyr::case_when(

    word_neg$item1 == "sever" ~ "sever(ity)",

    TRUE ~ as.character(word_neg$item1)
)

word_neg$item2<- dplyr::case_when(

    word_neg$item2 == "sever" ~ "sever(ity)",

    TRUE ~ as.character(word_neg$item2)
)

word_neg
```

```
## # A tibble: 264 x 3
##    item1     item2         n
##    <chr>     <chr>     <dbl>
##  1 hard      afraid        5
##  2 concern   problem        4
##  3 suspect   concern        3
##  4 concern   slow           3
##  5 slow      sever(ity)     3
##  6 suspect   problem        3
##  7 hard      concern        2
##  8 afraid    concern        2
##  9 overwhelm concern        2
## 10 weak      concern        2
## # ... with 254 more rows
```

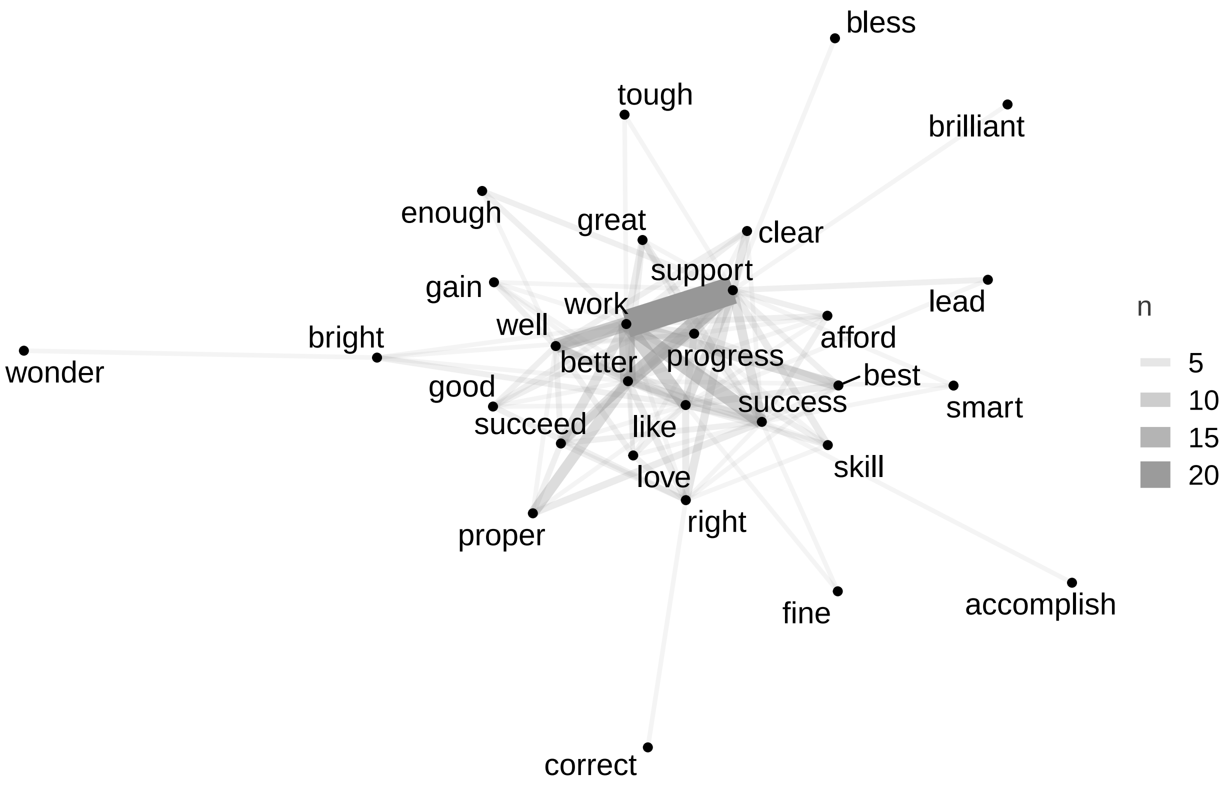
```
library(igraph)
library(ggraph)

set.seed(1234)
plot_cor_neg <- word_neg %>%
    filter(n >= 2) %>%
    graph_from_data_frame() %>%
    ggraph(layout = "fr") +
    geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "#979797") +
    geom_node_point(size = 1.5) +
    geom_node_text(aes(label = name),
                   size = 5,
                   repel = TRUE,
                   point.padding = unit(0.2, "lines")) +
    theme_void(base_size = 13) +
    theme(legend.text=element_text(size = 13),
          legend.title = element_text(size = 13, margin = margin(b = 5), color = "#3B3B3B"))


plot_cor_neg
```



```
word_pos <- dfm_td %>%
    inner_join(get_sentiments("bing")) %>%
    filter(sentiment == "positive") %>%
    pairwise_count(word, participant, sort = TRUE, upper = FALSE) %>%
    ungroup()

word_pos
```

```
## # A tibble: 375 x 3
##    item1   item2       n
##    <chr>   <chr>   <dbl>
##  1 work    support    21
##  2 work    better     11
##  3 better  support    11
##  4 work    success    10
##  5 work    well       10
##  6 work    like        9
##  7 better  proper      7
##  8 work    best        7
##  9 like    support     6
## 10 success support     6
## # ... with 365 more rows
```

```
set.seed(1234)
plot_cor_pos <- word_pos %>%
    filter(n >= 2) %>%
    graph_from_data_frame() %>%
    ggraph(layout = "fr") +
    geom_edge_link(aes(edge_alpha = n,
                       edge_width = n),
                   edge_colour = "#979797") +
    geom_node_point(size = 1.5) +
    geom_node_text(aes(label = name),
                   size = 5,
                   repel = TRUE,
                   point.padding = unit(0.2, "lines")) +
    theme_void(base_size = 13) +
    theme(legend.text=element_text(size = 13),
          legend.title = element_text(size = 13, margin = margin(b = 5), color = "#3B3B3B"))

plot_cor_pos
```


bless
tough
brilliant
enough
great
clear
gain
support
lead
work
well
bright
afford
better
progress
wonder
good
best
smart
success
succeed
like
skill
love
right
proper
fine
accomplish
correct
n