Topic 7: Word Embeddings

Joe DeCesaro

2022-05-17

This week's Rmd file here: https://github.com/MaRo406/EDS_231-text-sentiment/blob/main/topic_7.Rmd

Assignment

Download a set of pretrained vectors, GloVe, and explore them.

Grab data here:

Use the last three chunks of this markdown to produce the assignment.

```r
wiki_data <- read_table(file = here('data/glove/glove.6B.300d.txt'),
                        col_names = FALSE)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   X1 = col_character()
## )
## i Use `spec()` for the full column specifications.
```

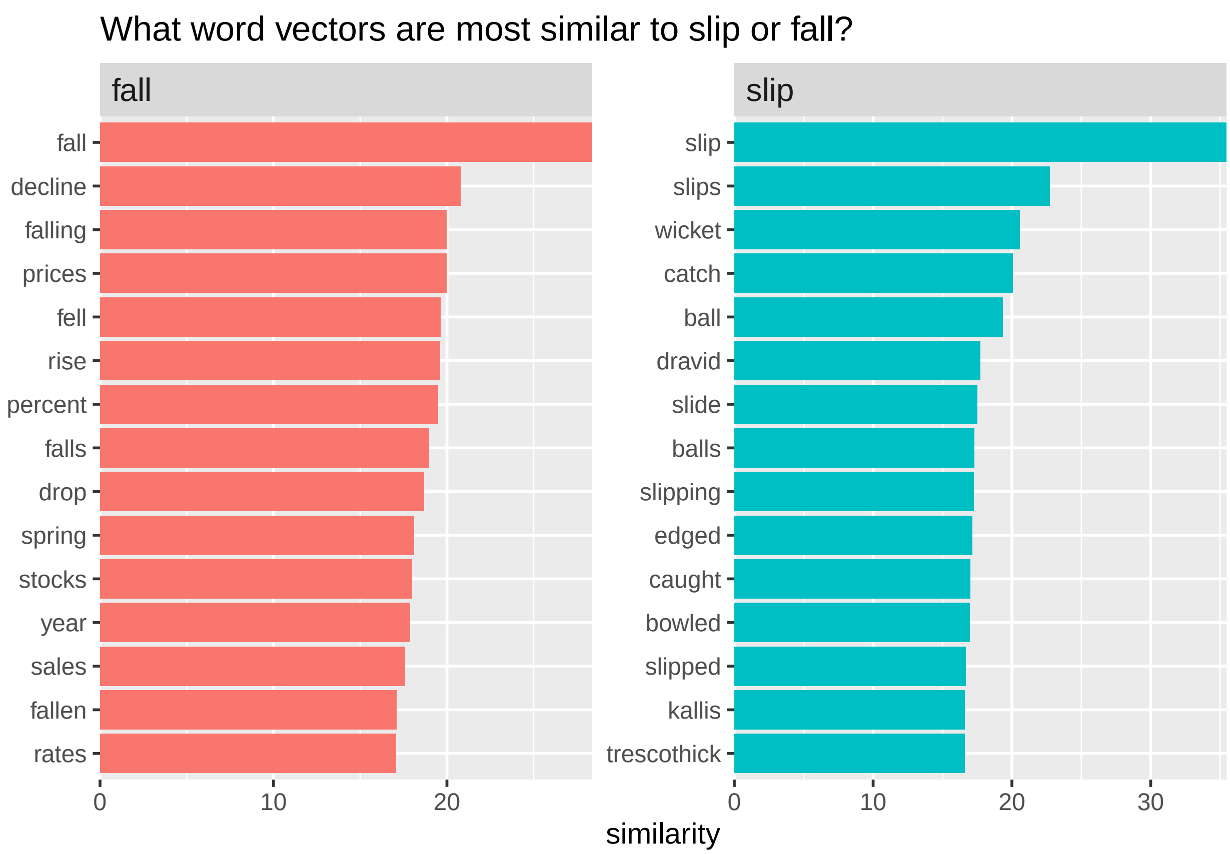
```r
wiki_data <- wiki_data %>%
  column_to_rownames(var = "X1")
#rownames(wiki_data) <- wiki_data$X1

word_vectors <- as.matrix(x = wiki_data)
```

```r
search_synonyms <- function(word_vectors, selected_vector) {
dat <- word_vectors %*% selected_vector

similarities <- dat %>%
        tibble(token = rownames(dat), similarity = dat[,1])

similarities %>%
       arrange(-similarity) %>%
        select(c(2,3))
}
```

1. Recreate the analyses in the last three chunks (find-synonyms, plot-synonyms, word-math) with the GloVe embeddings. How are they different from the embeddings created from the climbing accident data? Why do you think they are different?

```r
fall <- search_synonyms(word_vectors,word_vectors["fall",])
slip <- search_synonyms(word_vectors,word_vectors["slip",])
```

```r
slip %>%
    mutate(selected = "slip") %>%
    bind_rows(fall %>%
                  mutate(selected = "fall")) %>%
    group_by(selected) %>%
    top_n(15, similarity) %>%
    ungroup %>%
    mutate(token = reorder(token, similarity)) %>%
    ggplot(aes(token, similarity, fill = selected)) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~selected, scales = "free") +
    coord_flip() +
    theme(strip.text=element_text(hjust=0, size=12)) +
    scale_y_continuous(expand = c(0,0)) +
    labs(x = NULL, title = "What word vectors are most similar to slip or fall?")
```



These graphs vary wildly from the climbing incident data with words close to fall being much more associated with financial words or closer to the word itself like "falling". Slip also has much similar words, like "slips", but also seems to have a greater variety of similar words. We did not remove variations of words in this data so that is why we are getting slips, falling, and more. The climbing data set was for the sport so it makes sense that there are different word associations when compared to this data.

```r
# take semantics of snow and danger and see what happens when they are added together
snow_danger <- word_vectors["snow",] + word_vectors["danger",]
search_synonyms(word_vectors, snow_danger)
```

```
## # A tibble: 400,000 x 2
##    token      similarity
##    <chr>           <dbl>
##  1 snow             57.6
##  2 rain             40.6
##  3 danger           40.5
##  4 snowfall         34.8
##  5 weather          34.4
##  6 winds            34.0
##  7 rains            34.0
##  8 fog              33.6
##  9 landslides       33.3
## 10 threat           33.0
## # ... with 399,990 more rows
```

```r
# remove snow and association of snow from danger and see what happens
no_snow_danger <- word_vectors["danger",] - word_vectors["snow",]
search_synonyms(word_vectors, no_snow_danger)
```

```
## # A tibble: 400,000 x 2
##    token        similarity
##    <chr>             <dbl>
##  1 danger             23.3
##  2 risks              20.2
##  3 imminent           18.7
##  4 dangers            17.9
##  5 risk               17.8
##  6 32-team            17.6
##  7 mesdaq             17.5
##  8 inflationary       17.4
##  9 risking            17.2
## 10 2001-2011          17.0
## # ... with 399,990 more rows
```

Snow and danger together seems to have a lot more weather words than in the climbing data. When snow association is removed from danger it seems to focus on risk and some other, more random words.

2. Run the classic word math equation, “king” - “man” = ?

```r
no_king_man <- word_vectors["king",] - word_vectors["man",]
search_synonyms(word_vectors, no_king_man)
```

```
## # A tibble: 400,000 x 2
##    token       similarity
##    <chr>            <dbl>
##  1 king              35.3
##  2 kalākaua          26.8
```

```
##   3 adulyadej         26.3
##   4 bhumibol          25.9
##   5 ehrenkrantz       25.5
##   6 gyanendra         25.2
##   7 birendra          25.2
##   8 sigismund         25.1
##   9 letsie            24.7
##  10 mswati            24.0
## # ... with 399,990 more rows
```

We get a lot of words that are likely the word "king" in other languages or names of kings.

3. Think of three new word math equations. They can involve any words you'd like, whatever catches your interest.

```r
no_baseball_bat <- word_vectors["baseball",] - word_vectors["bat",]
search_synonyms(word_vectors, no_baseball_bat)
```

```
## # A tibble: 400,000 x 2
##    token        similarity
##    <chr>             <dbl>
##  1 baseball           31.0
##  2 basketball         30.1
##  3 football           26.5
##  4 nba                25.6
##  5 soccer             25.5
##  6 nfl                23.8
##  7 nhl                22.3
##  8 ncaa               22.3
##  9 hockey             22.2
## 10 professional       22.0
## # ... with 399,990 more rows
```

```r
no_surfing_wave <- word_vectors["surfing",] - word_vectors["wave",]
search_synonyms(word_vectors, no_surfing_wave)
```

```
## # A tibble: 400,000 x 2
##    token                 similarity
##    <chr>                      <dbl>
##  1 surfing                     34.1
##  2 windsurfing                 26.5
##  3 snorkeling                  26.1
##  4 http://thomas.loc.gov       24.7
##  5 snowboarding                24.3
##  6 kayaking                    24.3
##  7 http://www.boston.com       23.4
##  8 snorkelling                 23.1
##  9 biking                      22.9
## 10 skateboarding               22.4
## # ... with 399,990 more rows
```

```r
santa_barbara <- word_vectors["santa",] + word_vectors["barbara",]
search_synonyms(word_vectors, santa_barbara)
```

```
## # A tibble: 400,000 x 2
##    token      similarity
##    <chr>           <dbl>
##  1 santa            74.7
##  2 barbara          59.2
##  3 calif.           49.3
##  4 maria            44.8
##  5 monica           43.6
##  6 california       43.3
##  7 clara            42.5
##  8 san              42.0
##  9 ynez             41.3
## 10 clarita          39.0
## # ... with 399,990 more rows
```