assignment_6_word_embeddings

Marie Rivers

2022-05-11

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

Grab data here:

```
incidents_df<-read_csv("https://raw.githubusercontent.com/MaRo406/EDS_231-text-sentiment/825b159b6da4c7
```

```
glove_data <- fread(here("data", "glove.6B.300d.txt"), header = FALSE)
glove_df <- glove_data %>%
     remove_rownames() %>%
     column_to_rownames(var = 'V1')
```

First, let's calculate the unigram probabilities, how often we see each word in this corpus.

```
unigram_probs <- incidents_df %>%
    unnest_tokens(word, Text) %>%
    anti_join(stop_words, by = 'word') %>%
    count(word, sort = TRUE) %>%
    mutate(p = n / sum(n))
unigram_probs
```

```
## # A tibble: 25,205 x 3
##    word          n       p
##    <chr>     <int>   <dbl>
##  1 rope       5129 0.00922
##  2 feet       5101 0.00917
##  3 climbing   4755 0.00855
##  4 route      4357 0.00783
##  5 climbers   3611 0.00649
##  6 climb      3209 0.00577
##  7 fall       3168 0.00569
##  8 climber    2964 0.00533
##  9 rescue     2928 0.00526
## 10 source     2867 0.00515
## # ... with 25,195 more rows
```

Next, we need to know how often we find each word near each other word – the skipgram probabilities. This is where we use the sliding window.

```r
skipgrams <- incidents_df %>%
    unnest_tokens(ngram, Text, token = "ngrams", n = 5) %>%
    mutate(ngramID = row_number()) %>%
    tidyr::unite(skipgramID, ID, ngramID) %>%
    unnest_tokens(word, ngram) %>%
    anti_join(stop_words, by = 'word')
skipgrams
```

```
## # A tibble: 2,737,146 x 4
##    skipgramID `Accident Title`                          `Publication Y~` word
##    <chr>      <chr>                                                <dbl> <chr>
##  1 1_1        Failure of Rappel Setup (Protection Pulled~           1990 colo~
##  2 1_1        Failure of Rappel Setup (Protection Pulled~           1990 rocky
##  3 1_1        Failure of Rappel Setup (Protection Pulled~           1990 moun~
##  4 1_1        Failure of Rappel Setup (Protection Pulled~           1990 nati~
##  5 1_1        Failure of Rappel Setup (Protection Pulled~           1990 park
##  6 1_2        Failure of Rappel Setup (Protection Pulled~           1990 rocky
##  7 1_2        Failure of Rappel Setup (Protection Pulled~           1990 moun~
##  8 1_2        Failure of Rappel Setup (Protection Pulled~           1990 nati~
##  9 1_2        Failure of Rappel Setup (Protection Pulled~           1990 park
## 10 1_3        Failure of Rappel Setup (Protection Pulled~           1990 moun~
## # ... with 2,737,136 more rows
```

```r
#calculate probabilities
skipgram_probs <- skipgrams %>%
    pairwise_count(word, skipgramID, diag = TRUE, sort = TRUE) %>%
    mutate(p = n / sum(n))
```

```
## Warning: `distinct_()` was deprecated in dplyr 0.7.0.
## Please use `distinct()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

Having all the skipgram windows lets us calculate how often words together occur within a window, relative to their total occurrences in the data. We do this using the point-wise mutual information (PMI). It's the logarithm of the probability of finding two words together, normalized for the probability of finding each of the words alone. PMI tells us which words occur together more often than expected based on how often they occurred on their own.

```r
#normalize probabilities
normalized_prob <- skipgram_probs %>%
    filter(n > 20) %>%
    rename(word1 = item1, word2 = item2) %>%
    left_join(unigram_probs %>%
                  select(word1 = word, p1 = p),
              by = "word1") %>%
    left_join(unigram_probs %>%
                  select(word2 = word, p2 = p),
              by = "word2") %>%
    mutate(p_together = p / p1 / p2)
#Which words are most associated with "rope"?
```

```
normalized_prob %>%
    filter(word1 == "rope") %>%
    arrange(-p_together)
```

```
## # A tibble: 295 x 7
##    word1 word2        n          p      p1        p2 p_together
##    <chr> <chr>    <dbl>      <dbl>   <dbl>     <dbl>      <dbl>
##  1 rope  rope     25494 0.00340    0.00922 0.00922         40.0
##  2 rope  lengths    101 0.0000135  0.00922 0.0000575       25.4
##  3 rope  skinny      24 0.00000320 0.00922 0.0000144       24.2
##  4 rope  drag       211 0.0000281  0.00922 0.000138        22.1
##  5 rope  taut        98 0.0000131  0.00922 0.0000701       20.2
##  6 rope  coiled      60 0.00000800 0.00922 0.0000431       20.1
##  7 rope  thicker     21 0.00000280 0.00922 0.0000162       18.8
##  8 rope  trailing    68 0.00000907 0.00922 0.0000539       18.3
##  9 rope  fed         48 0.00000640 0.00922 0.0000413       16.8
## 10 rope  70m         31 0.00000414 0.00922 0.0000270       16.6
## # ... with 285 more rows
```

Now we convert to a matrix so we can use matrix factorization and reduce the dimensionality of the data.

```
pmi_matrix <- normalized_prob %>%
    mutate(pmi = log10(p_together)) %>%
    cast_sparse(word1, word2, pmi)

#remove missing data
pmi_matrix@x[is.na(pmi_matrix@x)] <- 0
#run SVD using irlba() which is good for sparse matrices
pmi_svd <- irlba(pmi_matrix, 100, maxit = 500) #Reducing to 100 dimensions
#next we output the word vectors:
word_vectors <- pmi_svd$u
rownames(word_vectors) <- rownames(pmi_matrix)
```

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?

```
search_synonyms <- function(word_vectors, selected_vector) {
dat <- word_vectors %*% selected_vector
# %*% is matrix multiplication

similarities <- dat %>%
        tibble(token = rownames(dat), similarity = dat[,1])
similarities %>%
       arrange(-similarity) %>%
        select(c(2,3))
}
```

```
glove_matrix <- as.matrix(glove_df)

fall_glove <- search_synonyms(glove_matrix, glove_matrix["fall",]) %>%
```

Table 1: Fall Synonyms

glove token	glove similarity	climb token	climb similarity
fall	28.35289	fall	0.1194042
decline	20.78131	rock	0.0523730
falling	19.97644	rope	0.0402845
prices	19.97596	line	0.0386850
fell	19.62625	short	0.0374310
rise	19.58406	ice	0.0353891
percent	19.46760	accident	0.0347391
falls	18.96819	foot	0.0344546
drop	18.66136	avalanche	0.0325806
spring	18.09208	coley	0.0324164
stocks	17.98144	gentzel	0.0322766
year	17.85333	lead	0.0321372
sales	17.56571	climber	0.0319858
fallen	17.08142	injuries	0.0311889
rates	17.06318	ground	0.0301290

```
  rename(glove_token = token) %>%
  rename(glove_similarity = similarity) %>%
  head(15)
slip_glove <- search_synonyms(glove_matrix, glove_matrix["slip",]) %>%
  rename(glove_token = token) %>%
  rename(glove_similarity = similarity) %>%
  head(15)
```

```
fall_climb <- search_synonyms(word_vectors,word_vectors["fall",]) %>%
  rename(climb_token = token) %>%
  rename(climb_similarity = similarity) %>%
  head(15)
slip_climb <- search_synonyms(word_vectors,word_vectors["slip",]) %>%
  rename(climb_token = token) %>%
  rename(climb_similarity = similarity) %>%
  head(15)
```

```
fall_synonyms <- cbind(fall_glove, fall_climb) %>%
  kable(col.names = c("glove token", "glove similarity", "climb token", "climb similarity"), caption =
fall_synonyms
```

```
slip_synonyms <- cbind(slip_glove, slip_climb) %>%
  kable(col.names = c("glove token", "glove similarity", "climb token", "climb similarity"), caption =
slip_synonyms
```

The glove generated synonyms for fall and slip seem more like true synonyms than those generated from the climbing incident data. This is likely due to the fact that the glove dataset is much larger and includes many more unique words. The climb synonyms for fall and slip seem to be words associated with the lead up and aftermath of the fall or slip (ie line, ice, injuries, fatal). All the top synonyms from the glove data have larger similarity scores than the climbing incident data.

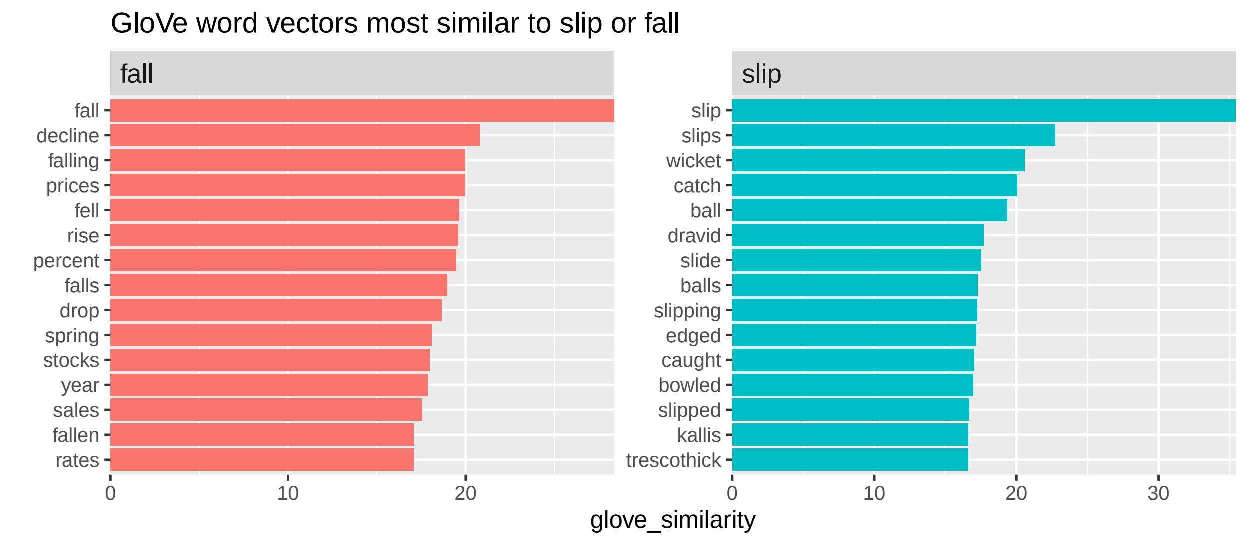
Table 2: Slip Synonyms

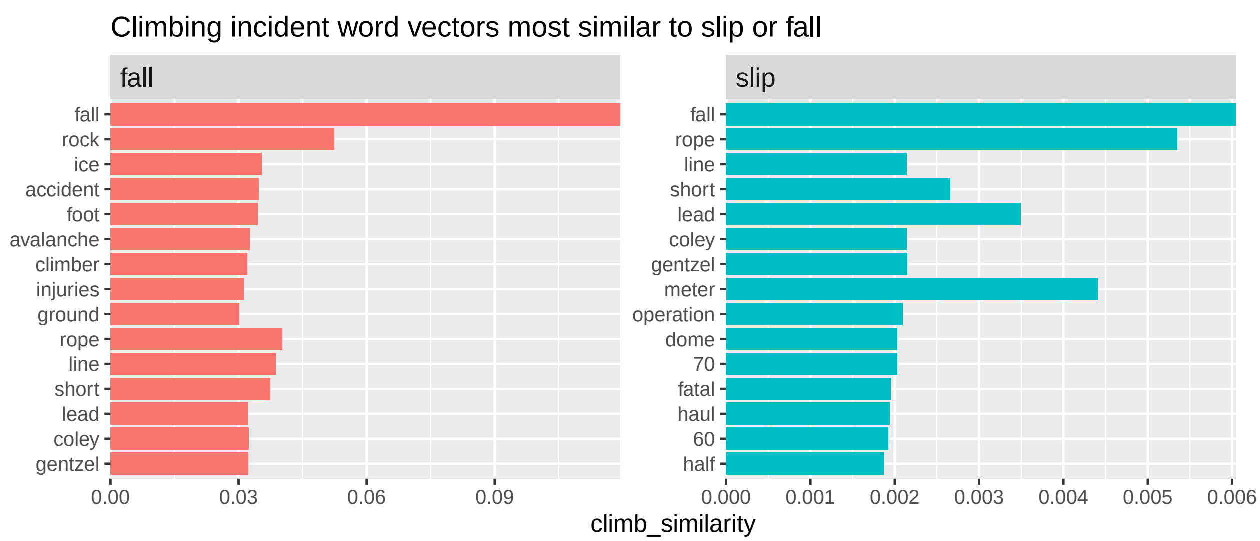
glove token	glove similarity	climb token	climb similarity
slip	35.43341	fall	0.0060439
slips	22.70521	rope	0.0053506
wicket	20.55729	meter	0.0044045
catch	20.05911	lead	0.0034921
ball	19.33358	short	0.0026588
dravid	17.70322	gentzel	0.0021469
slide	17.50436	coley	0.0021430
balls	17.26482	line	0.0021404
slipping	17.24516	operation	0.0020948
edged	17.14493	dome	0.0020325
caught	17.00399	70	0.0020283
bowled	16.95056	fatal	0.0019515
slipped	16.67810	haul	0.0019419
kallis	16.59541	60	0.0019252
trescothick	16.58964	half	0.0018700

```
glove_synonym_plot <- slip_glove %>%
    mutate(selected = "slip") %>%
    bind_rows(fall_glove %>%
                  mutate(selected = "fall")) %>%
    mutate(glove_token = reorder(glove_token, glove_similarity)) %>%
    ggplot(aes(glove_token, glove_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 = "GloVe word vectors most similar to slip or fall")

climb_synonym_plot <- slip_climb %>%
    mutate(selected = "slip") %>%
    bind_rows(fall_climb %>%
                  mutate(selected = "fall")) %>%
    mutate(climb_token = reorder(climb_token, climb_similarity)) %>%
    ggplot(aes(climb_token, climb_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 = "Climbing incident word vectors most similar to slip or fall")

synonym_plot <- glove_synonym_plot / climb_synonym_plot
synonym_plot
```






```r
snow_danger_climb <- word_vectors["snow",] + word_vectors["danger",]
search_synonyms(word_vectors, snow_danger_climb)
```

```
## # A tibble: 9,104 x 2
##    token      similarity
##    <chr>           <dbl>
##  1 snow           0.396
##  2 avalanche      0.131
##  3 conditions     0.0918
##  4 soft           0.0806
##  5 wet            0.0783
##  6 ice            0.0769
##  7 icy            0.0735
##  8 slope          0.0703
##  9 fresh          0.0604
## 10 blindness      0.0596
## # ... with 9,094 more rows
```

```r
no_snow_danger_climb <- word_vectors["danger",] - word_vectors["snow",]
search_synonyms(word_vectors, no_snow_danger_climb)
```

```
## # A tibble: 9,104 x 2
##    token     similarity
##    <chr>          <dbl>
##  1 avalanche     0.0882
##  2 danger        0.0547
##  3 rockfall      0.0540
##  4 gulch         0.0534
##  5 class         0.0507
##  6 hazard        0.0403
##  7 hazards       0.0394
##  8 occurred      0.0376
##  9 potential     0.0373
## 10 mph           0.0361
## # ... with 9,094 more rows
```

```r
snow_danger_glove <- glove_matrix["snow",] + glove_matrix["danger",]
search_synonyms(glove_matrix, snow_danger_glove)
```

```
## # 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
no_snow_danger_glove <- glove_matrix["danger",] - glove_matrix["snow",]
search_synonyms(glove_matrix, no_snow_danger_glove)
```

```
## # 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
```

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

```
king_man_glove <- glove_matrix["king",] - glove_matrix["man",]
search_synonyms(glove_matrix, king_man_glove) %>%
  head(15)
```

```
## # A tibble: 15 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
## 11 soopers           22.9
## 12 władysław         22.9
## 13 tuanku            22.8
## 14 prussia           22.7
## 15 norodom           22.6
```

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

```
lake_fish <- glove_matrix["lake",] + glove_matrix["fish",]
search_synonyms(glove_matrix, lake_fish) %>%
  head(15)
```

```
## # A tibble: 15 x 2
##    token      similarity
##    <chr>           <dbl>
##  1 lake             72.5
##  2 fish             68.2
##  3 trout            53.8
##  4 freshwater       52.3
##  5 river            51.5
##  6 lakes            51.0
##  7 salmon           49.7
##  8 water            49.1
##  9 fishing          49.0
## 10 pond             48.5
## 11 salt             47.0
## 12 species          45.0
## 13 creek            44.8
## 14 sea              43.7
## 15 ponds            41.8
```

```r
surf_no_ocean <- glove_matrix["surf",] - glove_matrix["ocean",]
search_synonyms(glove_matrix, surf_no_ocean) %>%
  head(15)
```

```
## # A tibble: 15 x 2
##    token         similarity
##    <chr>              <dbl>
##  1 surf                27.5
##  2 skateboarding       18.2
##  3 snowboard           17.9
##  4 surfing             17.5
##  5 feikens             17.3
##  6 zines               17.3
##  7 namfrel             16.8
##  8 snowboarding        16.8
##  9 andouille           16.6
## 10 punks               16.5
## 11 e-islam             16.5
## 12 longwell            16.4
## 13 desensitized        16.3
## 14 rockabilly          16.3
## 15 mc5                 16.3
```

```r
house_no_roof <- glove_matrix["house",] - glove_matrix["roof",]
search_synonyms(glove_matrix, house_no_roof) %>%
  head(15)
```

```
## # A tibble: 15 x 2
##    token         similarity
##    <chr>              <dbl>
##  1 house               29.3
##  2 senate              25.6
##  3 rep.                24.1
##  4 congressional       24.0
##  5 congress            23.6
##  6 clinton             23.4
##  7 republican          22.3
##  8 republicans         21.8
##  9 pelosi              21.8
## 10 steny               21.7
## 11 gingrich            21.6
## 12 democrats           21.6
## 13 democrat            20.6
## 14 parliament          20.6
## 15 democratic          20.2
```