Homework 6: Word Embeddings

Paloma Cartwright

2022-05-16

Classwork Data Set-up

```
incidents_df <- read_csv("https://raw.githubusercontent.com/MaRo406/EDS_231-text-sentiment/825b159b6da4
```

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.

```
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')

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
```

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

#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)
```

```
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)
```

Synonym Function

```
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))
}
```

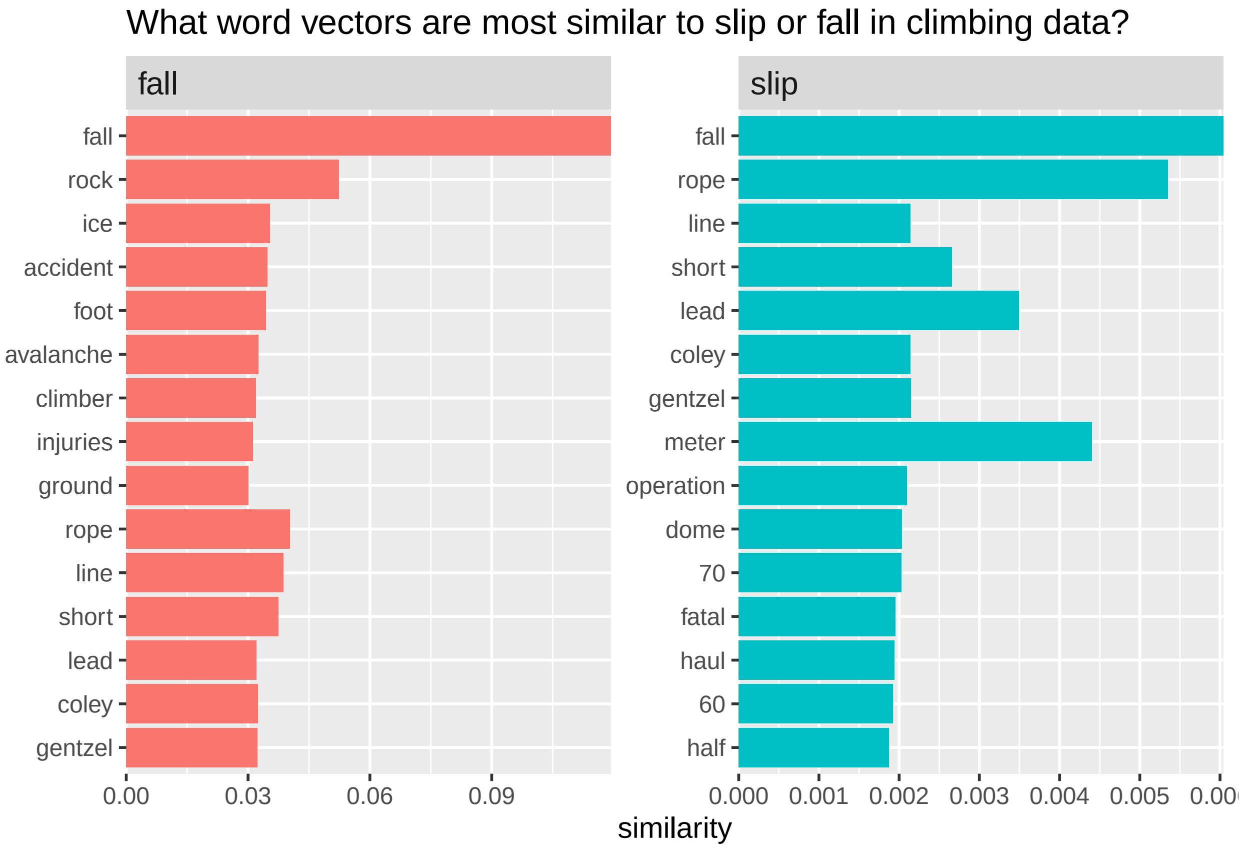
Find the synonyms in the climbing data

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

Plot the synonyms in the climbing data

```
climb_syn_plot <- slip_climb %>%
  mutate(selected = "slip") %>%
  bind_rows(fall_climb %>%
              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 in climbing data?")

climb_syn_plot
```



Word Math on the climbing data

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

```
## # 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
```

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

```
## # 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
```

Grab GloVe Data

```
# download.file('https://nlp.stanford.edu/data/glove.6B.zip', destfile = 'data/glove.6B.zip')
# unzip('data/glove.6B.zip')

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

Recreate the Analyses on GloVe data

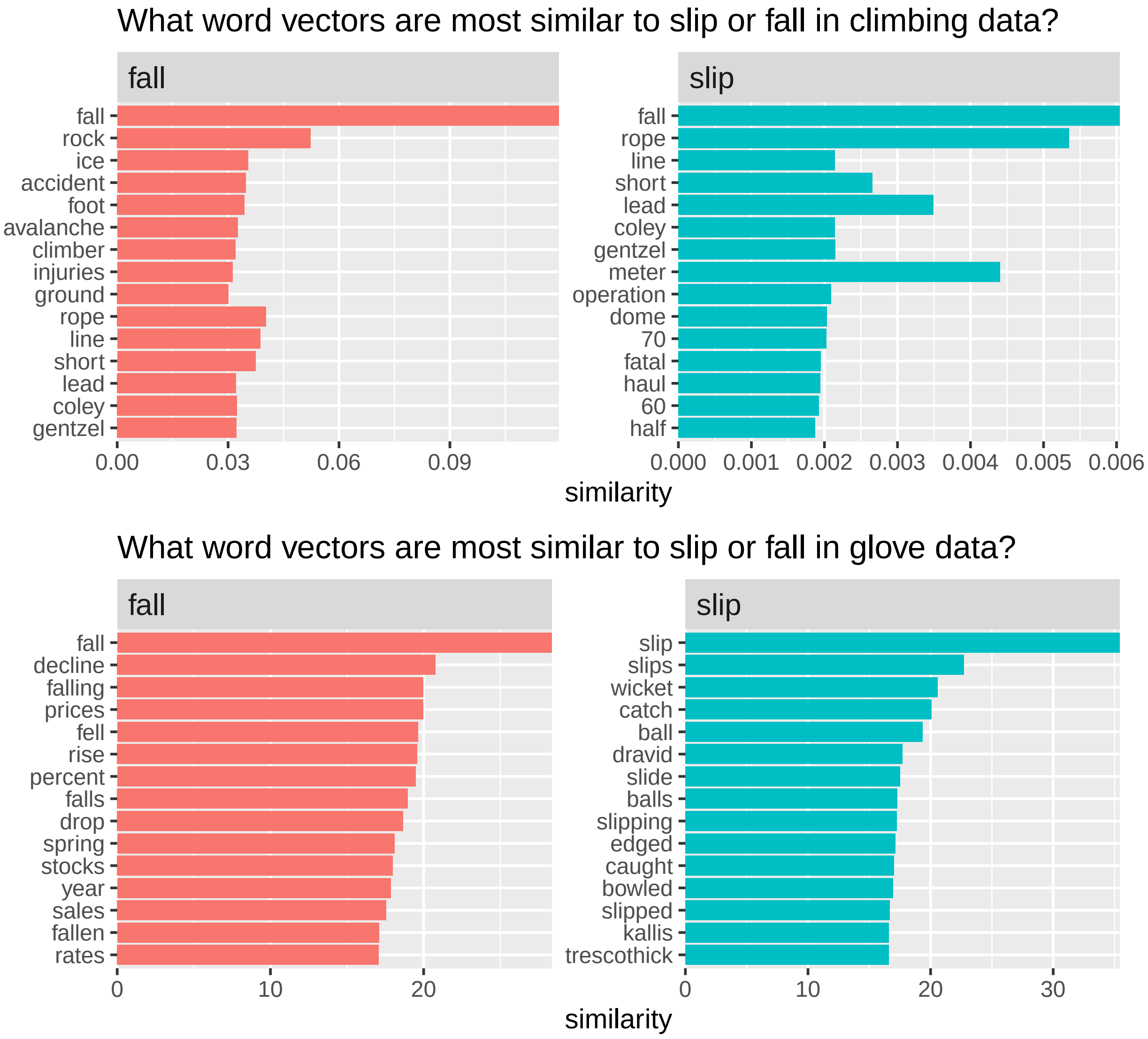
Find Synonyms in the glove data

How are they different from the embeddings created from the climbing accident data? Why do you think they are different?

```
glove_vectors <- as.matrix(glove_df)
fall_glove <- search_synonyms(glove_vectors, glove_vectors["fall",])
slip_glove <- search_synonyms(glove_vectors, glove_vectors["slip",])
```

```
glove_syn_plot <- slip_glove %>%
  mutate(selected = "slip") %>%
  bind_rows(fall_glove %>%
              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 in glove data?")
```

```
climb_syn_plot / glove_syn_plot
```



The similarity scores in the glove data are much higher than the similarities in the climbing data and the top words in each differ greatly. I think that's because the climbing data is very specific to climbing events but the glove data is much more broad so it covers a lot more varying topics.

Do Word Math on the Glove Data

```
snow_danger <- glove_vectors["snow",] + glove_vectors["danger",]
search_synonyms(glove_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
```

```
no_snow_danger <- glove_vectors["danger",] - glove_vectors["snow",]
search_synonyms(glove_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
```

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

```
king_man <- glove_vectors["king",] - glove_vectors["man",]
search_synonyms(glove_vectors, 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
```

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

```
summer_winter <- glove_vectors["summer",] + glove_vectors["winter",]
search_synonyms(glove_vectors, summer_winter)
```

```
## # A tibble: 400,000 x 2
##    token        similarity
##    <chr>             <dbl>
##  1 winter             80.5
##  2 summer             69.0
##  3 olympics           53.8
##  4 spring             51.1
##  5 season             49.1
##  6 autumn             47.9
##  7 temperatures       46.2
##  8 weather            46.1
##  9 universiade        45.0
## 10 paralympics        43.6
## # ... with 399,990 more rows
```

```
basketball_soccer <- glove_vectors["basketball",] - glove_vectors["soccer",]
search_synonyms(glove_vectors, basketball_soccer)
```

```
## # A tibble: 400,000 x 2
##    token      similarity
##    <chr>           <dbl>
##  1 celtics          20.3
##  2 lakers           18.1
##  3 3-point          17.8
##  4 pistons          17.3
##  5 3-pointers       17.2
##  6 3-pointer        16.8
##  7 pacers           16.7
##  8 knicks           16.5
##  9 76ers            16.4
## 10 rebounds         16.2
## # ... with 399,990 more rows
```

```
water_desert <- glove_vectors["water",] + glove_vectors["desert",]
search_synonyms(glove_vectors, water_desert)
```

```
## # A tibble: 400,000 x 2
##    token      similarity
##    <chr>           <dbl>
##  1 water            67.3
##  2 desert           64.2
##  3 sea              46.3
##  4 river            43.0
##  5 arid             42.7
##  6 dry              42.7
##  7 sand             41.7
##  8 soil             41.0
##  9 irrigation       40.1
## 10 lake             40.1
## # ... with 399,990 more rows
```