# Populate Keywords Database

Anthony Arroyo, Josh Forster

10/22/2022

## Datasets

We pulled a variety of job descriptions regarding data science jobs. Shown below are some the tweakable parameters to create the database.

```
postingsJson = "data/job_description_data.json"
dontWrite = TRUE

job = jsonlite::read_json(here::here(postingsJson))
jobIdVector = sapply(job, function(x) { x$link }) %>%
  magrittr::set_names(paste0("linkedin_", 1:length(.)), .)
writeFiles = rep("data/keywords_linkedin/", length(jobIdVector))

read.csv(here::here("data/keyword_posting_crosswalk.csv")) %>%
  rbind(., data.frame(job_id = jobIdVector, file_store = writeFiles, job_store = postingsJson, job_url =
    .[! duplicated(.), ] %>%
      write.csv(., here::here("data/keyword_posting_crosswalk.csv"), row.names = FALSE)


if (dontWrite) {
  writeFiles = NA
} else {
  writeFiles = here::here(writeFiles)
}

head(jobIdVector, 2)
```

```
## https://www.linkedin.com/jobs/view/data-scientist-%E2%80%93-operations-research-at-bosch-usa-3302078
##
##                                  https://www.linkedin.com/jobs/view/data-scientist-at-big-cloud-330939
##
```

## Captures from the English Dictionary

To determine our stop words, we used Webster's dictionary to select nouns, verbs, and adjectives.

```
captureGroups = c("n.", "a.", "v.")
dictionary = here::here("data/dictionary.json") %>%
```

```
  readLines(.) %>%
    jsonlite::fromJSON(.)

head(dictionary[-1:-705, -3])
```

```
##      pos        word                        synonyms
## 706   v.      ACCORD                            <NA>
## 707   a.  ACCORDABLE                            <NA>
## 708   n.  ACCORDANCE Harmony; unison; coincidence.
## 709   n.  ACCORDANCY                            <NA>
## 710   a.   ACCORDANT                            <NA>
## 711 adv. ACCORDANTLY                            <NA>
```

## Finished Aggregates

The data created shows the keyphrase, number of occurences across all inputs, and number of words. We can use this to determine the most common keyphrases in Data Science job descriptions.

```
values = GenerateKeywords(job, jobIdVector, writeFiles, dictionary, captureGroups, GrabLinkedin)
aggregates = SumFreq(values)
singularAggregates = aggregates %>% .[.$numWords == 1, ]
writeReports = FALSE

if (writeReports) {
  write.csv(aggregates, here::here("data/outputs/aggregateLinkedinPhrases.csv"), row.names = F)
  write.csv(singularAggregates, here::here("data/outputs/aggregateLinkedinKeywords.csv"), row.names = F)
}

head(aggregates[-1:-10, ], 20)
```

```
##             keyword sumFreq numWords
## 11             WORK      50        1
## 12     DATA SCIENCE      44        2
## 13 MACHINE LEARNING      42        2
## 14          MACHINE      41        1
## 15         ANALYSIS      39        1
## 16              ARE      39        1
## 17             WILL      37        1
## 18        ANALYTICS      35        1
## 19             TEAM      33        1
## 20           MODELS      31        1
## 21      ENGINEERING      30        1
## 22               BE      29        1
## 23         RESEARCH      28        1
## 24         MODELING      27        1
## 25              NEW      27        1
## 26           STRONG      27        1
## 27          ABILITY      26        1
## 28           OF THE      26        2
## 29      STATISTICAL      26        1
## 30       ABILITY TO      25        2
```

## Combining all data points

If we prefer to look at the data in summary as opposed to by source, that is available too. By providing each data folder, we are loading our database and can select all available sets. Furthermore, the available sets are made known by reading the crosswalk table containing the primary keys for the interop.

```
allValues = here::here("data/keyword_posting_crosswalk.csv") %>%
  read.csv(.) %>%
    .$file_store %>%
      unique(.) %>%
        here::here(.) %>%
          lapply(., LoadKeywordDatabase) %>%
            do.call(c, .)

allAggregates = SumFreq(allValues)
allSingularAggregates = allAggregates %>% .[.$numWords == 1, ]
writeFinalReports = FALSE

if (writeFinalReports) {
  write.csv(allAggregates, here::here("data/outputs/aggregateAllPhrases.csv"), row.names = F)
  write.csv(allSingularAggregates, here::here("data/outputs/aggregateAllKeywords.csv"), row.names = F)
}

if (tools::md5sum(here::here("data/outputs/aggregateAllPhrases.csv")) == "224a1868ac42be8e8390d273f142b
  allAggregates[c(16, 18, 21, 27, 31, 34, 42, 46, 50, 53, 54, 67, 115, 140, 178, 179), ]
} else {
  head(allAggregates[-1:-15, ], 20)
}
```

```
##                  keyword sumFreq numWords
## 16              ANALYSIS      86        1
## 18             ANALYTICS      76        1
## 21      MACHINE LEARNING      71        2
## 27                MODELS      60        1
## 31           STATISTICAL      56        1
## 34                PYTHON      53        1
## 42              RESEARCH      47        1
## 46            STATISTICS      43        1
## 50                   SQL      40        1
## 53            TECHNIQUES      39        1
## 54            ANALYTICAL      38        1
## 67         COMMUNICATION      35        1
## 115     COMPUTER SCIENCE      24        2
## 140        DATA ANALYSIS      21        2
## 178 COMMUNICATION SKILLS      18        2
## 179       DATA ANALYTICS      18        2
```