CS 410: Text Information Systems
University of Illinois at Urbana-Champaign
Fall 2022

Course Project – Progress Report

Submitted by: **Samuel Kirby**
slkirby2@illinois.edu

November 5, 2022

Project Track:	Free Topics		
Project Title:	Use of sentiment analysis on mutual fund quarterly commentaries		
Team Members:	Samuel Kirby	slkirby2	Solo project

(1) Progress Made Thus Far

Project Objective Refinements

One of the open questions in my project proposal was how to evaluate the success/accuracy of the model. After further thought on this subject I have considered three different potential approaches:

1) Using human-labelled commentaries. For the scope of this project, this would be both time-consuming and subject to bias, if a single individual determined the sentiment of commentary language. Ideally several people would perform such labelling in tandem.

2) Comparing sentiment scores to portfolio positioning. For example, if a fund manager has negative or "bearish" sentiment about the future, you would expect their portfolio to have a lower degree of Beta (market sensitivity) relative to its own history, or a comparison of peers.

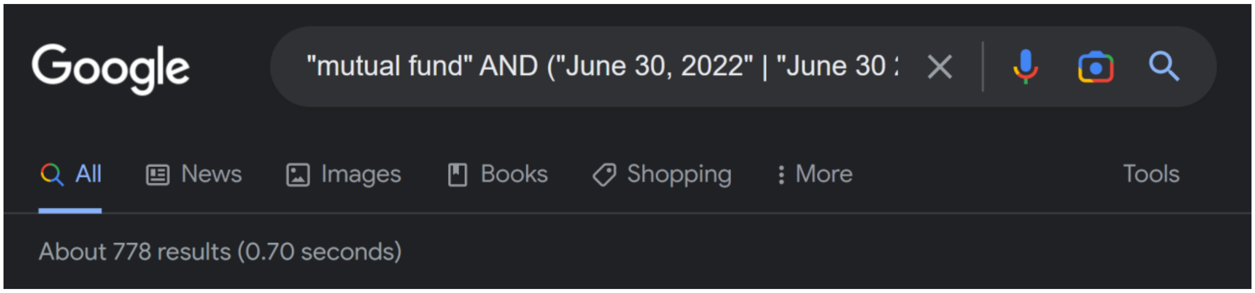
3) Comparing sentiment scores to what actually happened within the market in the subsequent quarter, and evaluating the performance of the fund relative to its peers or benchmark. This is conditioned upon the idea outlined in approach 2 – that a fund manager should position their portfolio based upon their sentiment – and uses actual market results to evaluate their success.

At present, I am leaning toward approach #3. Specifically, I intend to collect and analyze mutual fund commentaries for the quarter ending 2002-06-30, and evaluate results based upon the subsequent quarter ending 2022-09-30. Note that this approach will require some additional data collection and processing work such as extracting the correct portfolio benchmark for each manager, obtaining performance information for those benchmarks, obtaining performance for the fund, and perhaps most challenging, obtaining performance information for its peers within its investment category.

Data Collection

As I suspected in my project proposal, a properly structured and specific Google search query yielded excellent success in identifying PDF files of quarterly investment commentaries for mutual funds. The search query below, for example, resulted in 778 search results:

> *"mutual fund" AND ("September 30, 2022" | "Sept. 30, 2022" | "9/30/2022" | "3Q 2022" | "3Q2022" | "Sept 30 2022" | “2022-09-30”) AND "quarter" AND "performance" AND "contributors" AND "detractors" AND "outlook" AND "before investing" AND "past performance" filetype:pdf*



"mutual fund" AND ("June 30, 2022" | "June 30 :

About 778 results (0.70 seconds)

The next step was to systematically download the files returned by this search. I suspect that others have developed approaches to automated data collection from Google or other search engines, but I did not wish to invest project time in this particular area – especially since I would only perform this step a single time for this project. My approach was therefore to simply copy and paste each page of the Google search results into Microsoft Word (which maintained URL hyperlinks), save the resulting combined list of results as an HTML file, and then use the following R script to parse, extract, and perform basic cleansing of the list of links.

```
pg = read_html("2q2022_from_word.htm")
links = html_attr(html_nodes(pg, "a"), "href")
results = as.data.frame(links)

# remove duplicates
results = as.data.frame(results[!duplicated(results),])

# remove rows beginning with: https://www.google.com
results = results[!grepl("https://www.google.com", results[,1]),]
results = as.data.frame(results)
colnames(results) = c("URL")

# remove rows that do not contain "PDF" or "pdf"
pdf_results = dplyr::filter(results, grepl("PDF|pdf",URL))
```

From this list of URLs, I used the following code to attempt to download the PDF files and save them to a local directory, while handling errors/exceptions in cases where the file was not available.

```
path = "files/2Q22/"

for (f in 1:nrow(pdf_results)){

  es = paste("Downloading file #: ",f,sep="")
  print(es)

  # extract filename as the text following final "\" character in URL
  fn = strsplit(pdf_results[f,1], "/", perl=TRUE)[[1]]
  fn = fn[length(fn)]
  fp = paste(path,fn,sep="")

  tryCatch(download.file(pdf_results[f,1], fp, mode="wb"),
           error = function(e) print("Download error....skipping file."))

  Sys.sleep(5)
}
```

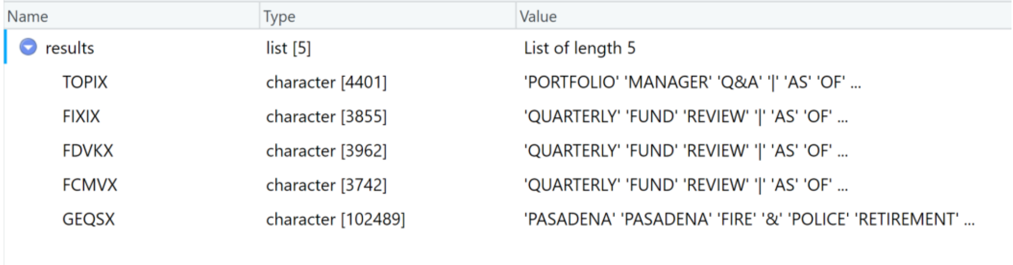
This resulted in a directory containing more than 220 PDF files. Most of these files meet the target criteria, however some manual curation will be required to eliminate files that are not representative of the target (such as mutual fund prospectuses, annual reports, semi-annual reports, marketing materials, etc.).

Data Extraction

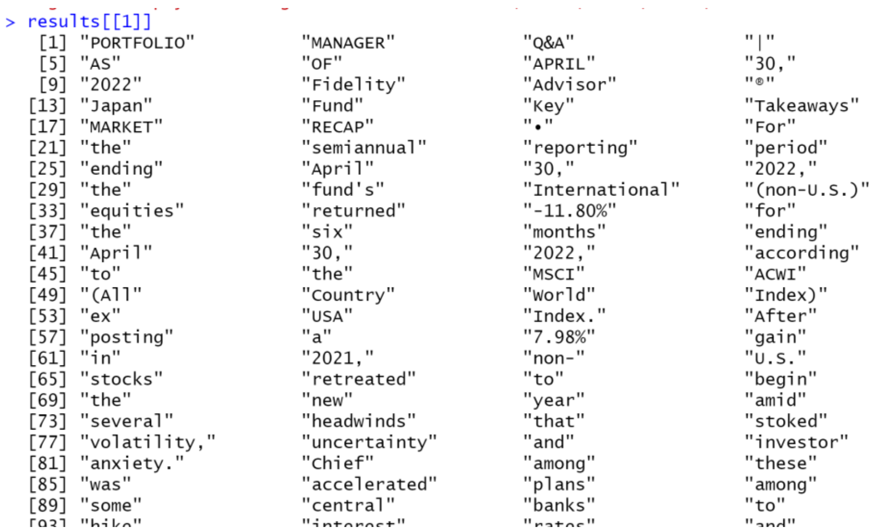
Another area of progress has been the extraction of text from PDF documents, using the “pdftools” library in R. The following R script iterates through the list of downloaded files (in this example, limited to 10 files rather than the entire directory), extract a vector of words from each “element” or region of the PDF file, combines these into a single list, locates a ticker symbol using regex if one exists (five-letter, all-cap sequence ending in “X”), uses this ticker to name the vector, and finally append the results file with the results of the current document. This approach was successful as shown below:

```
# initialize results list
mat = matrix(ncol=0, nrow=0)
results = list()


# get list of files within directory
```

```
path=getwd()

files = list.files(path = "files/2Q22")

# loop over files
for (f in 1:10){ #length(files)) {
  this_file = file.path("files/2Q22",files[f], fsep="/")
  print(this_file)
  file_contents = pdf_data(this_file)

  # file_contents is a nested list of X elements, where the 6th sub-element
"text" is a column of words.
  # loop over list and rbind these into a single column

  word_vec = data.frame(mat)
  file_words = data.frame(mat) # initialize for new file

  for (l in 1:length(file_contents)){
    word_vec = as.data.frame(file_contents[[l]][[6]])
    file_words = rbind(file_words, word_vec)
  }

  # check to see if a ticker symbol exists in this column (5 uppercase letters,
ending in X);
  # if so, make it the colname
  # if not, skip file and move to next

  ticker = str_extract(file_words, "[A-Z]{4}X+")
  print(ticker)

  if (is.na(ticker)==TRUE){
    next
  }

  colnames(file_words) = ticker
  results = append(results, file_words)
}
```



Name	Type	Value
results	list [5]	List of length 5
TOPIX	character [4401]	'PORTFOLIO' 'MANAGER' 'Q&A' '|' 'AS' 'OF' ...
FIXIX	character [3855]	'QUARTERLY' 'FUND' 'REVIEW' '|' 'AS' 'OF' ...
FDVKX	character [3962]	'QUARTERLY' 'FUND' 'REVIEW' '|' 'AS' 'OF' ...
FCMVX	character [3742]	'QUARTERLY' 'FUND' 'REVIEW' '|' 'AS' 'OF' ...
GEQSX	character [102489]	'PASADENA' 'PASADENA' 'FIRE' '&' 'POLICE' 'RETIREMENT' ...



```
> results[[1]]
  [1] "PORTFOLIO"        "MANAGER"          "Q&A"              "|"
  [5] "AS"               "OF"               "APRIL"            "30,"
  [9] "2022"             "Fidelity"         "Advisor"          "®"
 [13] "Japan"            "Fund"             "Key"              "Takeaways"
 [17] "MARKET"           "RECAP"            "•"                "For"
 [21] "the"              "semiannual"       "reporting"        "period"
 [25] "ending"           "April"            "30,"              "2022,"
 [29] "the"              "fund's"           "International"    "(non-U.S.)"
 [33] "equities"         "returned"         "-11.80%"          "for"
 [37] "the"              "six"              "months"           "ending"
 [41] "April"            "30,"              "2022,"            "according"
 [45] "to"               "the"              "MSCI"             "ACWI"
 [49] "(All"             "Country"          "World"            "Index)"
 [53] "ex"               "USA"              "Index."           "After"
 [57] "posting"          "a"                "7.98%"            "gain"
 [61] "in"               "2021,"            "non-"             "U.S."
 [65] "stocks"           "retreated"        "to"               "begin"
 [69] "the"              "new"              "year"             "amid"
 [73] "several"          "headwinds"        "that"             "stoked"
 [77] "volatility,"      "uncertainty"      "and"              "investor"
 [81] "anxiety."         "Chief"            "among"            "these"
 [85] "was"              "accelerated"      "plans"            "among"
 [89] "some"             "central"          "banks"            "to"
 [93] "hike"             "interest"         "rates"            "and"
```

(2) Remaining Tasks

- Manual curation of the downloaded PDF files to eliminate files that are not representative of the target (such as mutual fund prospectuses, annual reports, semi-annual reports, marketing materials, etc.).

- Evaluate additional data cleansing steps in an attempt to isolate the text that is more specific to the question of "outlook." For instance, trimming words before a specific keyword (such as "outlook", "forward", etc., as well trimming words at the end of the document that represent standard disclosures.

- Identify the model/algorithm to be used for unsupervised sentiment analysis using a vector of words. If I stay with this approach (which would be the most straightforward), the algorithm may be less available to extract meaning from text syntax and structure. To work around this limitation, I could attempt to "reconstitute" sentences from the vector of words. My concern, however, is that given the messy structure of text extracted from PDFs, the results of such reconstituted sentences could be less valuable.

- Perform sentiment analysis on the documents using at least one, and if possible, several different approaches/algorithms for comparison.

- Evaluate the results relative to benchmark-relative or peer group-relative performance in the following quarter, to assess how correct the fund manager's outlook was (assuming they incorporated this outlook into their portfolio management decisions).

(3) Any challenges/issues being faced

- Thus far I have been fortunate/lucky to have not faced material challenges in data collection. I expect more challenges to emerge as I move into the next phase of model/algorithm selection, implementation, and evaluation.

- I also expect to find challenges in extracting the benchmark/index for each fund in an automated way. I intend to try some simplistic pattern matching approaches or perhaps standardized formats embedded within required disclosure language. But, worst-case scenario, given my expected number of "surviving" documents following data extraction, the manual entry of benchmark data is also feasible with and expected 1-2 hours of work.

- Obtaining performance information for these indices and funds for the quarter ending 2022-09-30 for model evaluation purposes should be relatively straightforward using, for example, the Yahoo! Finance website or, ideally, freely-available APIs. However, obtaining performance information for the median fund within each respective peer group may pose a challenge using free/openly-available sources. If this is the case, I can fall back to using benchmark-relative performance as the primary evaluation mechanism – so I feel comfortable that I can work around this challenge if needed.