Conference Call Code Manual

Sixun Tang
Valerii Baidin

May 1, 2021

Abstract

This manual gives a detailed description of the whole process of exporting and processing conference calls for various textual analysis practices.

Contents

1 Overview 2

2 Data Structure 2

3 Step 1: Downloading Raw Data 3

4 Step 2: PDF Processing 7

5 Step 3: Firm Matching 8

1 Overview

This manual describes the whole process of exporting and processing conference calls for textual analysis from Thomson Streetevents database, which is available from Thomson One (Booth access). Thomson One updates latest conference calls on a regular basis and we need to rerun the codes to capture the latest call scripts. The main challenge is that the platform does not support scraping and poses a strict restriction on the maximum number of conference calls exported every time. As a result, we write automatic codes imitating the downloading action to circumvent the obstacle. Additionally, we use tools from Booth Mercury server and Julia codes to transfer the raw pdf files to usable text files stored in csv files. Finally, we match the conference calls with standard firm identifier, i.e., gvkey from Compustat, to facilitate linking to other firm data sets.

2 Data Structure

Currently we have already downloaded and processed conference calls from 01/08/2001 to 04/05/2021. The full data are stored in ConferenceCallData on Kilian's serve folder. The data are divided into five folders. The folders are organized by steps of the whole process, which we'll describe in details later.

1. **RawScripts**: Raw pdf files and OCR-transferred txt files of the original pdf files, i.e., each set of conference calls contain two files, one pdf and one txt.

2. **List**: xls files which contains the basic information of each conference call, including title, subtitle, report number, date, time, etc..

3. **CsvScripts**: Usable csv scripts. The csv file is processed by combining basic information from the .xls file and scripts from .txt file. These csv files could be directly fed to Python/Julia codes.

4. **FileRecord**: The records of conference calls in each time period (dta file). Additionally, there are two csv files containing the bad conference calls that could not be processed from txt files.We ran three rounds of collection and there were several bad conference calls in the first two round.

5. **FirmNameMatching**: csv files containing basic information of conference call and matched firm names from Compustat. The file could be linked to scripts using report number. For the first round, we have separate csv files for each year (2001-2020) and a total file of matched records with $\geq$ 90 matching probability. For round two and round three, the matching files are named as "*_202010" and "*_202104" respectively.

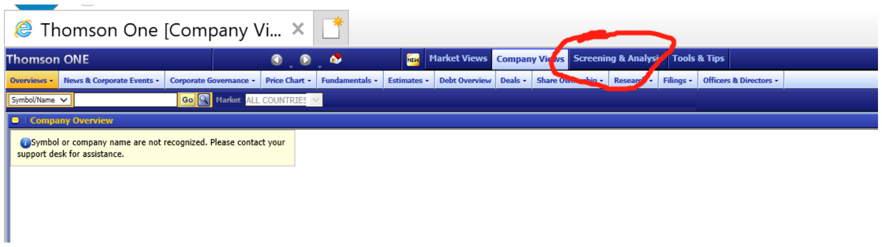
3 Step 1: Downloading Raw Data

There are several obstacles preventing us from exporting conference calls from the platform:

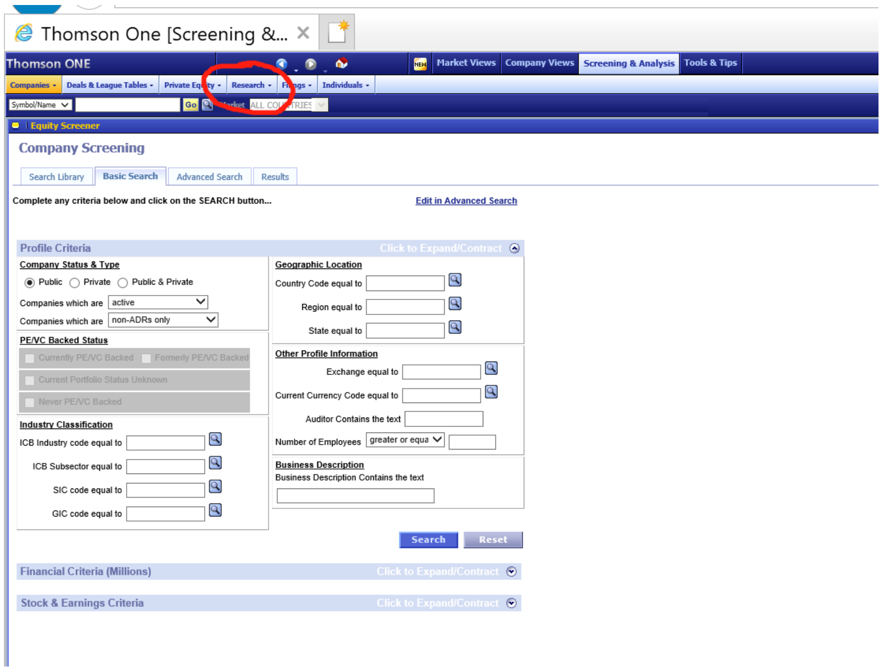
1. Each page will only show 50 conference calls.
2. A maximum of 2,000 conference calls will be presented for every search.
3. Web drivers are prohibited

To circumvent these obstacles, we design a series of codes to imitate the human downloading action. The summary of the procedure are shown in the following figures.

1. Use Internet Explorer (no other browers are allowed) to access Thomson One (`proxy.uchicago.edu/login/thomsonone`).
2. Go to *Screen Analysis-Research*



Symbol or company name are not recognized. Please contact your support desk for assistance.



Company Screening

Complete any criteria below and click on the SEARCH button...

Profile Criteria

Company Status & Type

[✓ Public] Private Public & Private

Companies which are: active

Companies which are: non-ADRs only

PE/VC Backed Status

Currently PE/VC Backed Formerly PE/VC Backed

Current Portfolio Status Unknown

Never PE/VC Backed

Industry Classification

ICB Industry code equal to

ICB Subsector equal to

SIC code equal to

GIC code equal to

Geographic Location

Country Code equal to

Region equal to

State equal to

Other Profile Information

Exchange equal to

Current Currency Code equal to

Auditor Contains the text

Number of Employees: greater or equa

Business Description

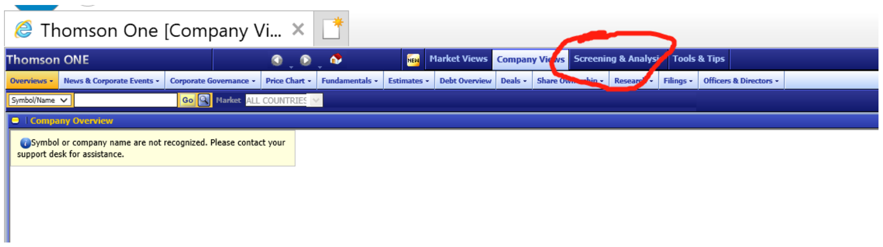
Business Description Contains the text

Financial Criteria (Millions)

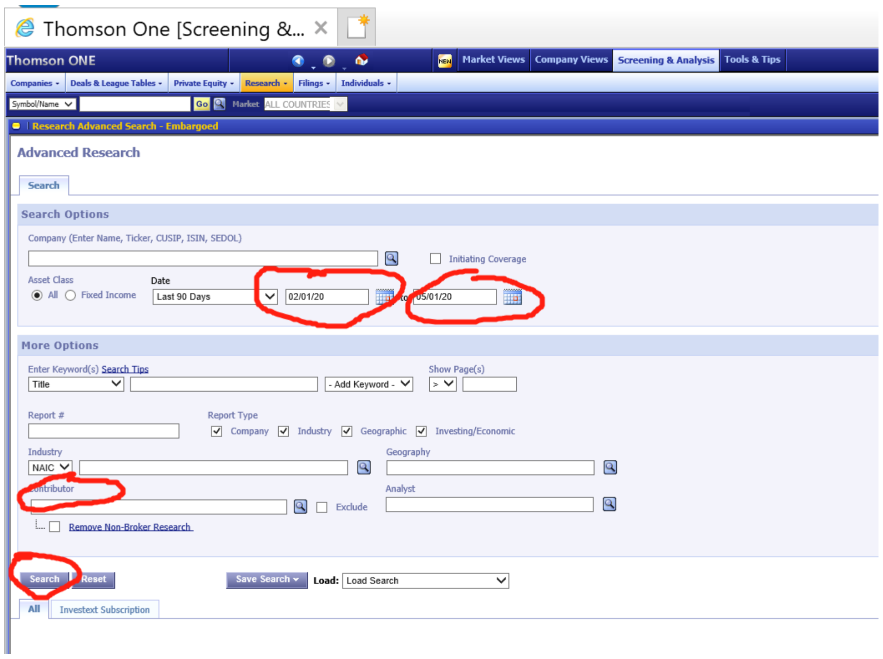
Stock & Earnings Criteria

3. Put *Streetevents* in the *Contributor* field and choose *Thomson Reuters Streetevents* in the drop-down list.

4. Enter in *Start Date* and *End Date* to decide the scope of conference calls. Then press *Enter*.



Symbol or company name are not recognized. Please contact your support desk for assistance.



Asset Class: [✓ All] Fixed Income

Date: Last 90 Days 02/01/20 to 05/01/20

Enter Keyword(s): Title

Show Page(s): >

Report Type: [✓ Company] [✓ Industry] [✓ Geographic] [✓ Investing/Economic]

Industry: NAIC

Load: Load Search

We use Python to download all the calls by using *pynput* package to automatically control the mouse and keyboard. The main code is automatic_download.py. mouse_key_recorder.py is a functional code that keeps record of the coordinates of positions that the mouse should click on. supplement_download.py takes a predetermined list as the input, and the downloading process is the same as the main automatic code.

The automatic code follows the steps below:

1. Enter start-time and end-time. We specify a four-day time period to ensure that each search gives no more than 2,000 calls. In round 2 and round 3 updates, since the number of conference calls is limited, we increase the time period.

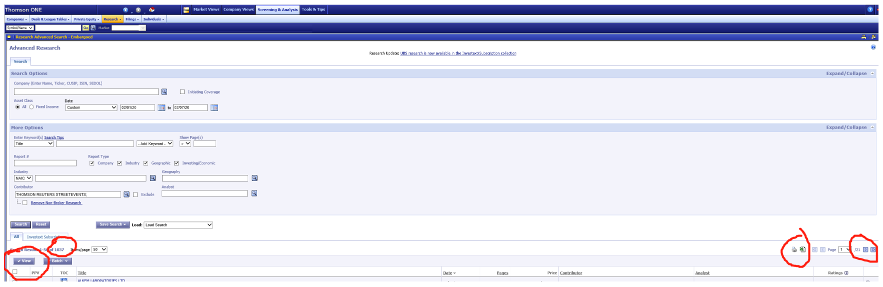
2. Copy the number of calls given by the search, save the number into a separate data set, and calculate the pages.

3. Select all calls in one page, download the pdf with all calls in that page, together with an excel which contains information of each call.[1] Also, to ensure that pdf and xls files are in pairs, we will check the existence of files in the folders. If there are some errors, the code will

[1]For each unit of observation in the excel, we have important variables including report number (unique identifier), report title (contains firm name) and date. This list is helpful in our parsing of the transformed pdf later on.

restart the process (by typing in the login websites, and automatically finishing the access steps. This is also the error handle step for any other errors).

4. Uncheck all calls, click next page, repeat Step 2 and Step 3. If this is the last page of the date period, back to Step 1 with date moving backwards.[2]



The automatic download is always affected by the network condition, and also Thomson One will automatically log off from time to time. So we need to handle the errors listed below:

1. Automatically log off.

2. Broken pdf pages and unsuccessful excel download.

3. Sudden network error.

4. Authentication error of the system (failure to log in).

5. The change of file orders when we re-enter the system. (when the time-off is long, e.g., several hours)

For most errors, the automatic code will try to re-enter the system by finishing all the access steps and retrieving the download from the previous break point. For error 4, please stop the code and wait for hours before we could access the system again. For error 5, we download the pdf first and then the excel because the pdf takes longer to download while the excel is quick to get. However, if the error occurs when the pdf is exported without the excel, we suggest to delete the pdf (when the error occurs, there will be one unpaired pdf) and re-download it to ensure the files are paired (which is important for the later processing).

[2]For all the mouse positions, we manually do one loop and use mouse_key_recorder.py to return a list of useful mouse positions.

If you run the code on a new laptop, please make sure to change the parameters below in the code:

- Make sure your computer is zoomed 100% in the settings.
- Use the mouse key recorder to capture the coordinates of every position and put it in the automatic code. This means that you should manually click on the positions to finish the step and the code will record the position. The coordinates will be accurate only if your computer is zoomed 100%.
- Set up start date and end date, as well as the time window.
- Set up the folders for the pdf and xls separately.

The output of this step are pdf files in the RawScripts, and xls files in the List.

4 Step 2: PDF Processing

The next big step is to process all the pdfs into usable data files. Note that one pdf contains many firms' calls, so another challenge is how to split one file into different calls. To achieve this, we use code ParseCCpdf_v1.jl and take the following steps:

1. Use the *pdftotext* tool from Mercury server to transfer pdfs (.pdf) to texts (.txt) file.[3] This tool would add page and paragraph delimiters while parsing the original file, which brings convenience for the split of firms. You could directly use pdftransfer.sh to submit jobs on the server. Please make sure you change the directory to the place where all raw pdfs are located.

2. Use Julia to split the texts file and construct a primary database which contains report information (in the downloaded excel files) and raw call scripts. Specifically, start from the delimiters contained in the .txt file to identify pages, then use *title* and *pages* in the information file to locate the beginning and end of each conference call.[4] Generate a new variable in the information file to store the raw call scripts. The primary database is a collection of .csv files with the following important variables:

[3] *pdftotext* is an OCR-reading tool. Information could be found at `https://poppler.freedesktop.org/`.

[4] For some transcripts, the texts are laid out in two columns in the same page. And during this process there are always small problems to be dealt with.

- **Title** (firm name), **Subtitle** (firm name, date, and whether final/primary transcripts)
- **Date**, **Pages** (the number of pages of the call)
- **Analyst** (analysts who collect these transcripts, different analysts have slightly different forms of transcripts.)
- **Report** (Unique report number)
- **Call** (Raw call transcripts)

The output of this step are txt files in the RawScripts, and csv files in the CsvScripts.

5 Step 3: Firm Matching

Another important part is to match firms with Compustat to get: i. GVKEY (unique firm identifier); ii. country information. The task isn't trivial since the name of firms doesn't have strict spelling for various reasons.

WRDS gives full assess to "Compustat - Capital IQ" database, which contains various datasets. We can search for data by using the web interface page or through SAS-Studio web application. SAS-Studio allows to have access to all raw data tables, which we are using by the following steps:

1. Open the SAS-studio web application: https://wrds-www.wharton.upenn.edu/pages/data/sas-studio-wrds/
2. All files are located at "Server Files and Folder" (to the right hand).
3. Captiliq auxiliary files are located at *wrds/capitaliq/sasdata/helper*.
4. Each table can be download by clicking the right mouse button. Although, it's not good practice, due to high size of tables.
5. The best way to download data is to create Query (right mouse button on a table ⇒ new ⇒ Query).
6. Downloading Query's result is a little bit tricky, since you can only print the result. The result is located in user's temporary folder. To find the name of the temporary folder: 1)

open any table 2)push "display the code .." button 3) run the code 4) Result window $\rightarrow$ Engine/Host Dependent Information $\rightarrow$ filename shows your temporary folder.

We used the following tables with specific columns from Compustat:

1. **CIQCOMPANY**: companyid, companyname, tickersymbol, countryid and other columns. Total: 24,511,757 obs. Public Companies: 66,256. Private Companies: 16,544,322. Public Investment Firms: 1,987 and Private Investment Firms: 203,407

2. **WRDS_GVKEY**: companyid, gvkey (115,357 observations).

3. **CIQCOUNTRY**: countryid, countryname: countryid, countryname (221 countries).

In addition, we have Hassan's Firm-Level Political Risk Dataset,[5] which also provides us with GVKEY and firm names. We think that this data set contains almost all the company's names that we have, since we use the same data source.[6]

We merged all tables into *compustat_company.csv.* This file contains *company name, compnany id, gvkey, tickersymbol, countryid and countryname.* The data from Hassan are named as FirmSVkey.csv. These files are in AuxData inside firmMatchingCode. To match firms' names with Compustat, we now have three sources: Hassan's dataset, Compustat dataset, and unique Tickers. If several firms has the same ticker, we don't use the ticker on the "perfect matched" step. We use program code linkCCtoGvkey.jl and follow the steps described below:

1. Process firms' names: 1) lower alphabets. 2) delete all special symbols. 3) delete all common words.[7]

2. Match firms by three sources[8] and combine all results. This is "Perfect Match" step, since we haven't used any guess method yet.

3. For the remaining unmatched firms, we use compare-function, which is based on Jaro string distance and return the probability that the two string are equal. First, we match firms with

[5]Firm-Level Risk website.

[6]In addition, in contrast to the Compustat dataset, Hassan's dataset has English spelling for all company's names, which helps us, since Conference Calls also has English spelling names.

[7]We deleted the words: group, plc, ltd, limited, ag, corp, corporation, Incorporation, laboratories, labs, the, holdings, oyj, inc, conference call, conference, co, company, trust, investment, investments, sln, sa, event transcript of, event brief of., and others.

[8]The result table has three additional columns: *gvkey_t, gvkey_h, gvkey_c* which are for Ticker, Hassan and Compustat datasets matching.

the same ticker (38% of a subset) and keep the best guess of a name if the probability is higher than 80%.[9] After that, the firm is matched with all possible names from Compustat and Hassan's dataset.[10] We can sure about matched name with probability above 90%.

The output of this step are the firm matching csvs in FirmNameMatching.

[9]We have checked that 80% is enough for matching names with the same ticker.

[10]Matched name and probability is saved into columns *gues_name and prob.*