**Contributions:**

* Attribution theory dictionary and reading engine, similar to LM, but with more complex semantic structures that can be reapplied to any research relating to internal, external or performance based 10-K vocabulary, or any financial use to extract components of 10-Ks that contain the aforementioned language
* Test the correlation between performance and internal and external sentiments
* Goodness of fit R value => when the year goes bad the correlation weakens
* Gets literature review condensed

**Online code database which does the following:**

* Formalize heuristics for regex extraction of a specific part of 10-K eg. MD&A
* In depth discussion of web scrapping efficiency
* Discussion of optimal method to optimize text extraction, parsing of large quantity of text data speed (writing to list, numpy arrays, using itertools)
* Data structure to maximize storage of text data
* Systematic way to select testing set (cik)

**10-K filing complete extraction**

**Extract raw files:**

extracthtml.py

convertcsv.py

DAR0.py

Missing: parsing heuristics but introduced in LM’s codes

**Using LM database:**

renamefiles.py

movenewdirs.py

movefiles.py

**MD&A extraction**

**Original testing concepts/trials:**

For loop + Boolean trails: tfidftrial1.py, test 2.py, extractsections.py

**Extract MD&A sample codes:**

onlinecodes.py / MDA.py

edgar1.py

**Finalized:**

newMDA.py, wordcount<100.py

**Item 13 extraction:**

convertcsv1.py

Item13.py

sortfailv4.py

extractpreequitysection.py

**Vocabulary**

tfidfvectorizer.py

getridofhightfidf.py

**Parser**

senttockenized.py

regexmatchwords.py

dicpart1.py

dicpart3v2.py

dicpart2.py

Reference LM’s code: Generic\_Parser.py

Others: LM stopwords.py

**POS tags**

transformwords.py

**Sorting words with heuristics**

analyze.py

aaronsortwords.py

useless.py

replacecountryname.py

removecurrenciesnumbers.py

getridoffutures.py

analyze2.py

alignwords.py

alignwords2.py

getridofperf.py

regexmatchwords.py

BERT1.py

BERT.py

I am extremely disorganized but I figured out that these are the key pieces of codes (if you skim through this you will get an idea of the whole progression of the project).

The parser bit needs optimizing.

The labeling files are:

intextfinal.xlsx : current progress of Aaron on labelling word

ridwords1.csv: words that Aaron got rid of

uselesswords1.xlsx: labelled useless words

final4.csv: final document I output after getting rid of useless words and stop words

We will continue to label words in final4.csv (but we need to get rid of the ones Aaron already labeled in the intextfinal.xlsx file)) I am finding the heuristics and Aaron is doing the labelling. James is helping me look through the performance dictionary I labeled. Aaron also has a friend who is potentially helping.

If you want to do some part of speech tagging over the internal and external dictionary here's some notes:

<https://docs.google.com/document/d/1w_dtDH09NyL_0rhCPmo30C4_SstM-4GkHTvMYwogwmI/edit?fbclid=IwAR3PrDY8qlJ5oSM1kYcoGIC4u9QPLc4wKDqiiqyxiSoXMrgVAg4h2172rTs>

If you want to look at some machine learning maths, I put a Bishop book in the google drive, it’s a bit tedious to go through, but it’s a classic to read.

* Due to economic fluctuations, company experienced negative earnings