2019

Text Based Signals - Midpoint Report

APPLIED FINANCE PROJECT

RUSTEM SHINKARUK

Contents

Introduction:

The purpose of this report to share our mid report analysis of the project. The original scope involves carrying out textual analysis from social media platforms namely Facebook, Twitter, YouTube and Amazon for any potential alpha through prediction of returns or forecasting of fundamentals of companies. The purpose of this analysis is to predict the behavior of returns of individual companies based on the general sentiment which people exhibit through their reviews. It is assumed that if the consensus over a product is relatively positive then the company's expected to perform better in the future. This drills down to the basic economics and supply demand relationship in the overall market. Often, people generally buy products that have good quality reviews compared to products that have negative or no reviews. Therefore, intuitively, this criterion is a natural indicator of the future sales of the companies and as we know, if the future sales of the companies are expected to increase, it will increase the overall net income and earnings of the company ultimately increasing the stock prices. Our goal in this initial analysis is to evaluate, clean and analyze the data with the possibility of trying different algorithms to map the products with their respective company and find out if there is any correlation between the our sentiment score, generated for each company from reviews and products database, and fundamentals of companies.

Literature review:

The literature around text-based signals seems to evolve day by day and has an upward slope especially the last decade when more and more researchers seem to be interested in the ways machine learning and the continuously increasing computational power can be used for acquiring novel results. The factors unveiled throughout the literature, seem to rely on a relatively small number of data sources and more emphasis seems to be given in ways to increase accuracy and "alpha" of algorithms like natural languages processing, rather than actually acquiring novel datasets. These would be datasets besides the Dow Jones, 10K's, Wall Street Journal etc.

The papers that we have analyzed so far are some of the most famous papers in the text-based signals literature, and there are plenty more to be explored. More specifically, these papers are "More Than Words: Quantifying Language to Measure Firms' Fundamentals" by Tetlock, et. al 2008 and "Lazy Prices" by Cohen et. al

"More Than Words: Quantifying Language to Measure Firms' Fundamentals" by Tetlock, et. al 2008

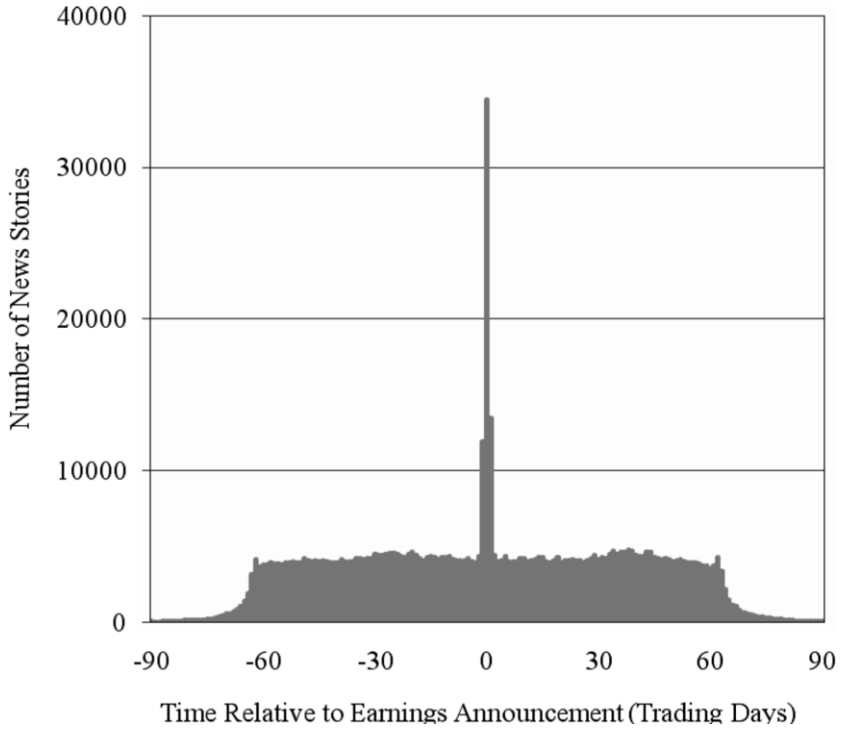
The basic idea behind this paper seems fairly simple: one has to measure the negative and positive words that appear in an article and thus extract the sentiment that it would create to a reader. This "feeling" could then be depicted at the readers trading decisions and as a result, in the market in general. The data used for this paper come for the Wall Street Journal and more specifically the column "Abreast of the Market" which gives daily news about the market. It is known that the market seems to overreact at negative news and underreact at positive. Couls this be the case in the word usage too? Indeed, the paper shows that the fraction of negative words in firm-specific news stories forecasts low firm earnings. The method, as we discussed is to count the negative words in the article. But before counting instances of negative words, they combine all qualifying news stories for each firm on a given trading day into a single composite story. Then they standardize the fraction of negative words in each composite news story by

subtracting the prior year's mean and dividing by the prior year's standard deviation of the fraction of negative words. In terms of mathematical formulas we could write:

$$Neg = \frac{\text{No. of negative words}}{\text{No. of total words}}$$

$$neg = \frac{Neg - \mu_{Neg}}{\sigma_{Neg}},$$

Where neg is the standardization of Neg. In the beginning, the authors try to confirm whether news stories concentrate around earnings announcement days. As shown in the figure below, the three adjacent spikes representing the firm-specific news stories one day before, on the same day as, and one day after a firm's earnings announcement. This finding suggests that news stories could play an important role in communicating and disseminating information about firms' fundamentals.

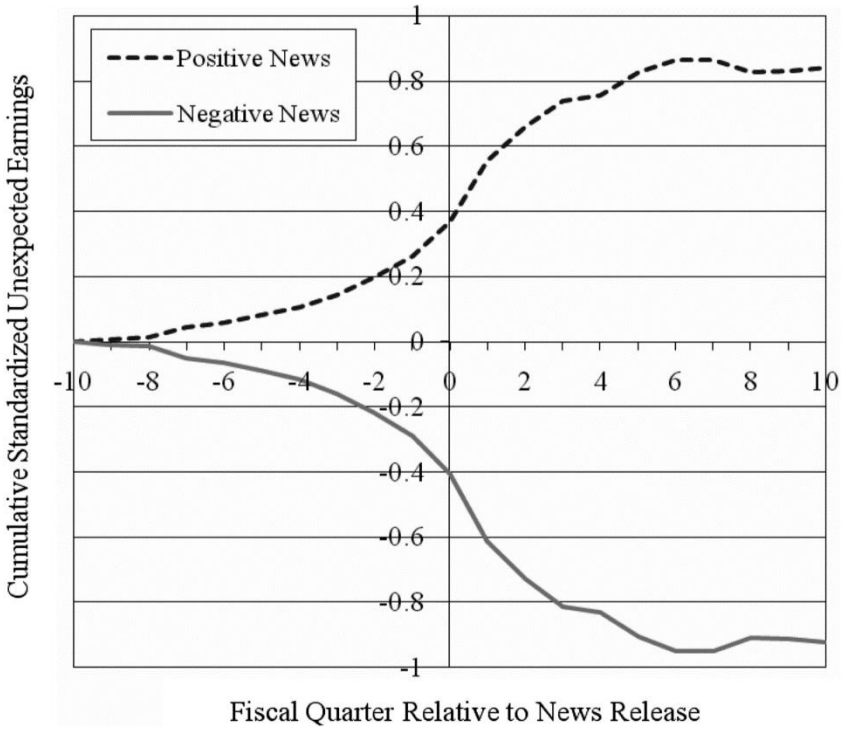


In the paper, authors use two measures of firms' quarterly accounting earnings as dependent variables in their predictability tests, as the quarterly frequency is the highest frequency for earnings data. Their main tests compute each firm's standardized unexpected earnings (SUE) which is defined as:

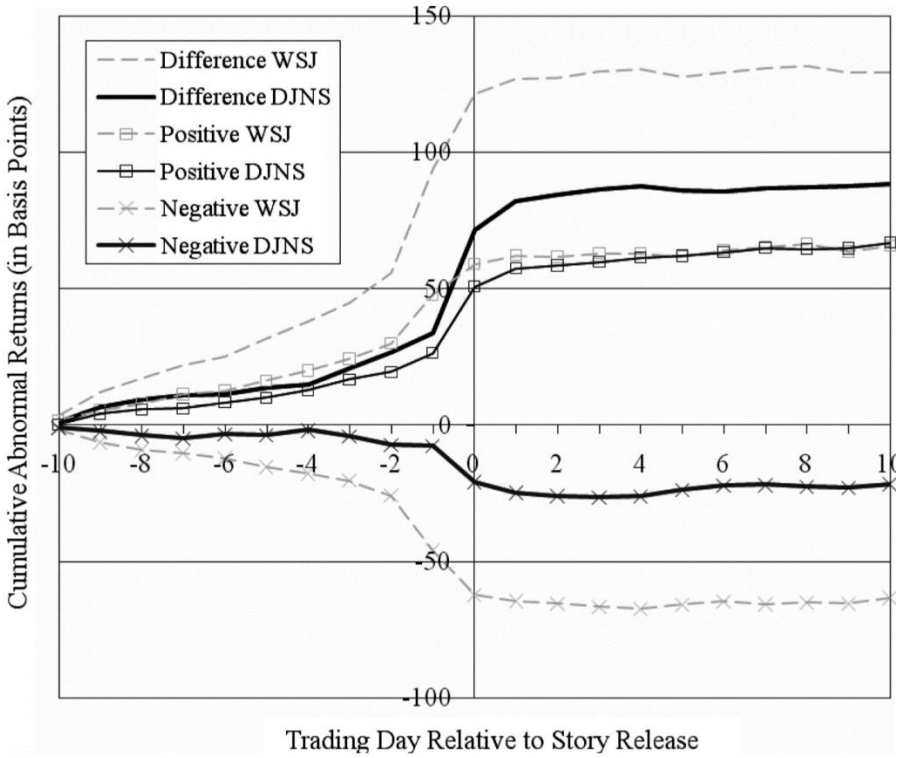
$$UE_t = E_t - E_{t-4}$$

$$SUE_t = \frac{UE_t - \mu_{UE_t}}{\sigma_{UE_t}},$$

Then the paper proceeds with computing firms' cumulative standardized unexpected earnings (SUE) from 10 fiscal quarters preceding media coverage of an earnings announcement to 10 quarters after the media coverage. They define media coverage of the announcement as positive (negative) when it contains a fraction of negative words in the previous year's top (bottom) quartile, where the measure of negative words is the fraction of words that are negative in the news stories from 30 trading days prior up to 3 trading days prior to an earnings announcement. They separately analyze the firms with positive and negative media coverage prior to their earnings announcements and compute the cumulative SUE for both sets of firms, beginning 10 quarters prior to the news and ending 10 quarters after the news. To compute SUE values after the news stories, only unexpected earnings benchmarks known at the time of the news were used. The results can be seen in the figure below:



This figure, shows that firms with negative news stories before an earnings announcement experience large negative shocks to their earnings that endure for at least 4 quarters after the news, hence they conclude that negative words in firm-specific stories leading up to earnings announcements significantly contribute to a useful measure of firms' fundamentals. To justify this result, one view is that this result is surprising because numerous stock analysts and investors closely monitor the actions of S&P500 firms. An alternative view is that negative words are informative measures of firms' fundamentals because they do not suffer from the same shortcomings as the quantitative variables that one can use to forecast earnings. Keeping this founding as a base for the rest of their research, authors then examine the market's apparently sluggish reaction to negative words in the 4 weeks surrounding the story's release to the public and they come up with the figure below.



The graph presents a firm's abnormal event returns from 10 trading days preceding a news story's release to 10 trading days following its release. The Fama-French three-factor model with a [−252, −31] trading day estimation period relative to the release of the news story was used as a benchmark. It can be understood that although the market reacts quite efficiently to positive and negative news, there is some delayed reaction, particularly for the DJNS news stories. Finally, they estimate the impact of reasonable transaction costs on the trading strategy's profitability. To judge the sensitivity of profits to trading costs, they recalculate the trading strategy returns under the assumption that a trader must incur a round-trip transaction cost of between zero and 10 bps. Table IV which can be seen below, displays the abnormal and raw annualized cumulative news-based strategy returns under these cost assumptions. From the evidence in Table IV, the authors conclude that the simple news-based trading strategy explored here is no longer profitable after accounting for reasonable levels of transaction costs.

Trading Costs (bps)	Abnormal Annualized Returns (%)	Raw Annualized Returns (%)
0	23.17	21.07
1	20.30	18.25
2	17.50	15.49
3	14.76	12.80
4	12.09	10.17
5	9.47	7.60
6	6.92	5.09
7	4.43	2.64
8	1.99	0.25
9	−0.39	−2.09
10	−2.71	−4.37

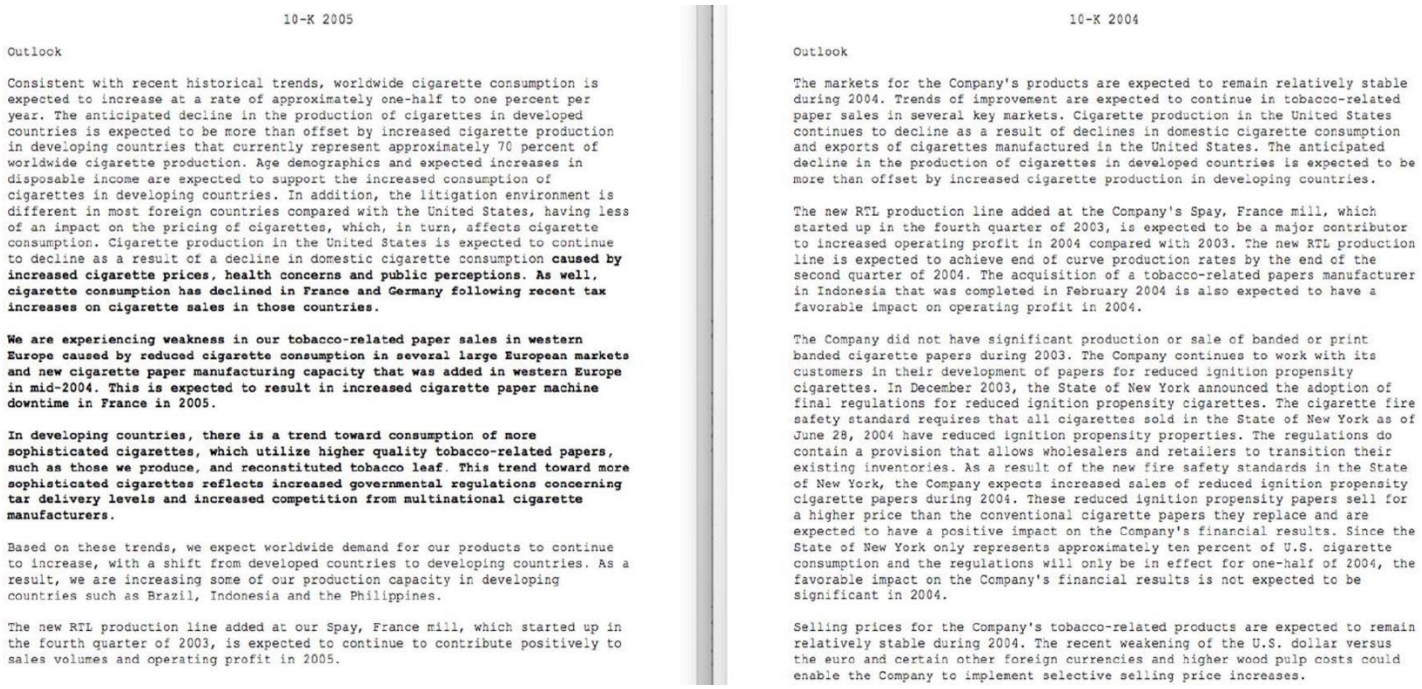
In conclusion, this paper comes up with two main significant points: that negative words in the financial press forecast low firm earnings and that that stock market prices incorporate the information embedded in negative words with a slight delay. Moreover, they conclude that the stock market is relatively efficient with respect to firms' hard-to-quantify fundamentals but they do find that market prices consistently underreact to negative words in firm-specific news stories, especially those that relate to fundamentals. Lastly, they underline that future research on quantifying language has the potential to improve our understanding of how information is incorporated in asset prices.

"Lazy Prices", Cohen et. al 2015

In this paper, Cohen, Malloy and Ngyun argue that investors are inattentive to certain information in quarterly or annual reports. In particular, firms seem to repeat almost all information in report, from their last reports. Using EDGAR data from 1994 to 2014, they show that active changes to the wording is associated with negative alpha of up to 22% per annum. They point out that the reporting changes are concentrated in the MD&A section, and more specifically in the management discussion. They conclude that changes in language referring to executive team such as CEO of CFO, or regarding litigation, are especially informative. They focus on the behavior of corporations, and show that when firms break from former language or well-codified text in their annual and quarterly reports, there is a substantial amount of information embedded in this action for important future firm outcomes. explore the implications of default choices in an entirely non experimental setting, by analyzing the behavior of corporations. More specifically, they focus on a setting where defaults appear to be commonly used by firms, namely

in their reporting decisions and show that the particular construction of firms' annual and quarterly reports suggests that firms are using simple default choices, the most obvious of which is simply repeating the information that they previously reported to the markets.

To prove their point, the authors provide two 10-K reports from two consecutive years (2004-2005). The part in bold is the only part that has undergone some kind of changes in terms of wording. According to the authors this is where our attention should be focused on, and this is the part that can be exploited to return significant results.



10-K 2005

Outlook

Consistent with recent historical trends, worldwide cigarette consumption is expected to increase at a rate of approximately one-half to one percent per year. The anticipated decline in the production of cigarettes in developed countries is expected to be more than offset by increased cigarette production in developing countries that currently represent approximately 70 percent of worldwide cigarette production. Age demographics and expected increases in disposable income are expected to support the increased consumption of cigarettes in developing countries. In addition, the litigation environment is different in most foreign countries compared with the United States, having less of an impact on the pricing of cigarettes, which, in turn, affects cigarette consumption. Cigarette production in the United States is expected to continue to decline as a result of a decline in domestic cigarette consumption **caused by increased cigarette prices, health concerns and public perceptions. As well, cigarette consumption has declined in France and Germany following recent tax increases on cigarette sales in those countries.**

We are experiencing weakness in our tobacco-related paper sales in western Europe caused by reduced cigarette consumption in several large European markets and new cigarette paper manufacturing capacity that was added in western Europe in mid-2004. This is expected to result in increased cigarette paper machine downtime in France in 2005.

In developing countries, there is a trend toward consumption of more sophisticated cigarettes, which utilize higher quality tobacco-related papers, such as those we produce, and reconstituted tobacco leaf. This trend toward more sophisticated cigarettes reflects increased governmental regulations concerning tar delivery levels and increased competition from multinational cigarette manufacturers.

Based on these trends, we expect worldwide demand for our products to continue to increase, with a shift from developed countries to developing countries. As a result, we are increasing some of our production capacity in developing countries such as Brazil, Indonesia and the Philippines.

The new RTL production line added at our Spay, France mill, which started up in the fourth quarter of 2003, is expected to continue to contribute positively to sales volumes and operating profit in 2005.

10-K 2004

Outlook

The markets for the Company's products are expected to remain relatively stable during 2004. Trends of improvement are expected to continue in tobacco-related paper sales in several key markets. Cigarette production in the United States continues to decline as a result of declines in domestic cigarette consumption and exports of cigarettes manufactured in the United States. The anticipated decline in the production of cigarettes in developed countries is expected to be more than offset by increased cigarette production in developing countries.

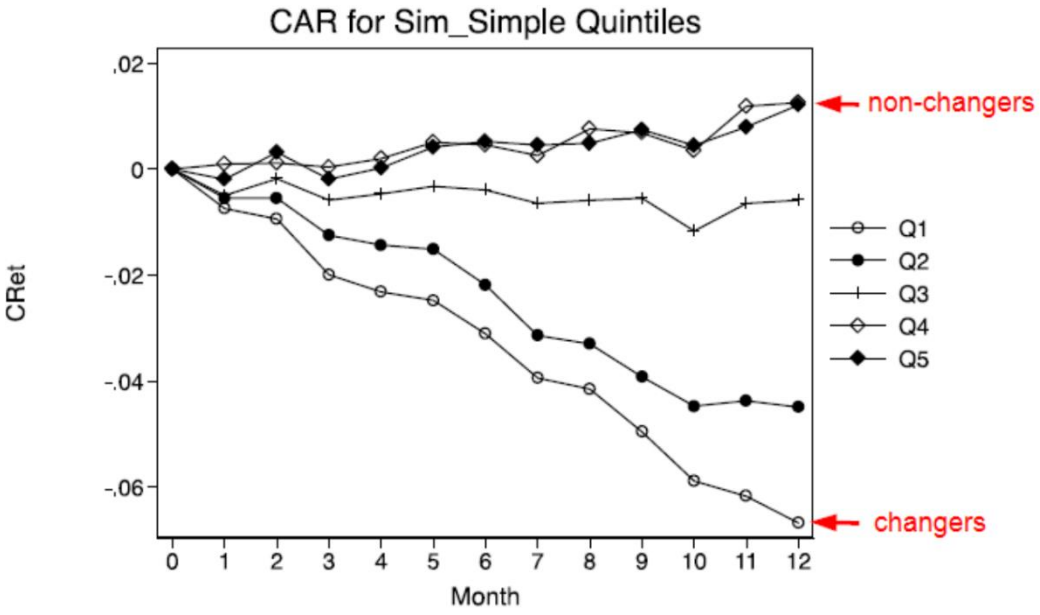
The new RTL production line added at the Company's Spay, France mill, which started up in the fourth quarter of 2003, is expected to be a major contributor to increased operating profit in 2004 compared with 2003. The new RTL production line is expected to achieve end of curve production rates by the end of the second quarter of 2004. The acquisition of a tobacco-related papers manufacturer in Indonesia that was completed in February 2004 is also expected to have a favorable impact on operating profit in 2004.

The Company did not have significant production or sale of banded or print banded cigarette papers during 2003. The Company continues to work with its customers in their development of papers for reduced ignition propensity cigarettes. In December 2003, the State of New York announced the adoption of final regulations for reduced ignition propensity cigarettes. The cigarette fire safety standard requires that all cigarettes sold in the State of New York as of June 28, 2004 have reduced ignition propensity properties. The regulations do contain a provision that allows wholesalers and retailers to transition their existing inventories. As a result of the new fire safety standards in the State of New York, the Company expects increased sales of reduced ignition propensity cigarette papers during 2004. These reduced ignition propensity papers sell for a higher price than the conventional cigarette papers they replace and are expected to have a positive impact on the Company's financial results. Since the State of New York only represents approximately ten percent of U.S. cigarette consumption and the regulations will only be in effect for one-half of 2004, the favorable impact on the Company's financial results is not expected to be significant in 2004.

Selling prices for the Company's tobacco-related products are expected to remain relatively stable during 2004. The recent weakening of the U.S. dollar versus the euro and certain other foreign currencies and higher wood pulp costs could enable the Company to implement selective selling price increases.

More specifically, regarding their methodology, the authors collect all complete 10-K, 10-K405, 10-KSB, and 10-Q filings from the SEC's database from 1994 to 2014. Then they measure the **quarter-on-quarter similarities** between 10-Q and 10-K filings using four different similarity measures using language processing software: 1) cosine similarity, 2) Jaccard similarity, 3) minimum edit distance, and 4) simple similarity. The authors then sort the firms into five quintiles: Quintile 1 (Q1) refers to firms that have the least similarity between their documents this year and those from last year; the authors refer to these companies as "**Changers.**" Quintile 5 (Q5) represents firms whose docs are most similar to those that came before; they refer to this group as "**Non-Changers.**" Firms are held in the portfolio for 3 months. Portfolios are rebalanced

monthly. They compare the future stock returns of “changers” to “Non-changers”. The results show that “changers” are associated with lower future returns. In particular, a portfolio that goes long “non-changers” and short “changers” in annual and quarterly financial reports earns a statistically significant **30-60 bps per month — equivalent to up to 7.6% over the following year.** The below figure shows the average cumulative abnormal return (CAR) for each quintile portfolio after portfolio formation.



It’s notable that the stock prices of the changers, which represent firms that have significantly altered their reporting, seem to suffer lasting effects that do not reverse. The authors dig deeper into the effect and find that filing changes are largely concentrated in the management discussion (MD&A) section, which facilitates the most flexibility in terms of content. However, text changes in the “Risk Factors” section, and the languages referring to CEO/CFO team, litigation and lawsuits are more informative for stock prices. In our mechanism tests, they measure the sentiment of document changes by counting the number of positive words minus the number of negative words in the changes between the old document and the new document, normalized by the size of the changes. They further compute the uncertainty and litigious nature of the change by counting the number of words categorized as uncertainty and litigious,

respectively, normalized by the size of the changes. Moreover, they examine the implications of firms' decisions to change the language and construction of their SEC filings. Their hypothesis is that large changes in reporting, when they do occur, will have significant implications for firms' future actions and outcomes, given the tendency of firms to simply report what they previously reported (i.e., to not change their reports).The authors begin by analyzing the future stock returns associated with firms who change their reports, versus those who do not. First they compute standard calendar-time portfolios, and then they control for additional determinants of returns by employing Fama-MacBeth monthly cross-sectional regressions.

Then the authors are focused on exploring the mechanism at work behind our key return results. They try to achieve this by regressing similarity measures on a host of characteristics of the documents in question. The goal of this exercise is to better understand what helps explain decreases in similarity across years for a given document and right after, they try to isolate the particular sections of the quarterly and annual reports that are associated with the largest declines in similarity across years for a given firm. They then take the item/section categories and examine the return predictability associated with changes to each section. Lastly, authors perform a series of robustness checks to ensure that our key findings are not simply repackaging a set of previously known return predictors. To do so, they re-run the Fama-MacBeth regressions, but include a series of additional firm-level characteristics. Collectively, findings indicate that these subtle changes in firms' reporting behavior have substantial predictability for future returns in a manner that has not previously been documented in the literature.

Overall, this paper does a good job showing that there are abnormal returns associated with strategies that use the changes in the template language in financial filings as a trading signal. This suggests that important information in financial filings is slowly incorporated in to prices. Under the semi-strong efficient market hypothesis, this should not happen but in the behavioral world that we live in, this is not surprising.

Preliminary Analysis:

Data Overview:

The scope of the project includes analysis on the “Reviews and Products Database” of Amazon for the period from May 1996 to July 2014 containing more than 142 million observations.

The goal is to create a large “Technology” database consisting of a number of subcategories and products sold through Amazon. To achieve this, we used from the available data resource the subcategories in the form of K-cores (i.e., dense subsets). These data have been reduced to extract the k-core, such that each of the remaining users and items have k reviews each. For our case, the subsets were 5-core, meaning in these datasets we only have products that have at least 5 reviews and users that have done at least 5 reviews. That allowed us to have a first filtering of our data, cutting out products that have no reviews and where no sentiment could be extracted and also excluding potential fake accounts created only to give negative comments to specific products and then remain inactive. In order to create our “Technology” dataset we combined the following 5-core subsets:

1) Electronics

2) Movies and TV

3) CDs and Vinyl

4) Cell Phones and Accessories

5) Video Games

6) Digital Music

All the raw data 5-core subcategories. The ones used for our dataset circled.

Books	5-core (8,898,041 reviews)	ratings only (22,507,155 ratings)
Electronics	5-core (1,689,188 reviews)	ratings only (7,824,482 ratings)
Movies and TV	5-core (1,697,533 reviews)	ratings only (4,607,047 ratings)
CDs and Vinyl	5-core (1,097,592 reviews)	ratings only (3,749,004 ratings)
Clothing, Shoes and Jewelry	5-core (278,677 reviews)	ratings only (5,748,920 ratings)
Home and Kitchen	5-core (551,682 reviews)	ratings only (4,253,926 ratings)
Kindle Store	5-core (982,619 reviews)	ratings only (3,205,467 ratings)
Sports and Outdoors	5-core (296,337 reviews)	ratings only (3,268,695 ratings)
Cell Phones and Accessories	5-core (194,439 reviews)	ratings only (3,447,249 ratings)
Health and Personal Care	5-core (346,355 reviews)	ratings only (2,982,326 ratings)
Toys and Games	5-core (167,597 reviews)	ratings only (2,252,771 ratings)
Video Games	5-core (231,780 reviews)	ratings only (1,324,753 ratings)
Tools and Home Improvement	5-core (134,476 reviews)	ratings only (1,926,047 ratings)
Beauty	5-core (198,502 reviews)	ratings only (2,023,070 ratings)
Apps for Android	5-core (752,937 reviews)	ratings only (2,638,172 ratings)
Office Products	5-core (53,258 reviews)	ratings only (1,243,186 ratings)
Pet Supplies	5-core (157,836 reviews)	ratings only (1,235,316 ratings)
Automotive	5-core (20,473 reviews)	ratings only (1,373,768 ratings)
Grocery and Gourmet Food	5-core (151,254 reviews)	ratings only (1,297,156 ratings)
Patio, Lawn and Garden	5-core (13,272 reviews)	ratings only (993,490 ratings)
Baby	5-core (160,792 reviews)	ratings only (915,446 ratings)
Digital Music	5-core (64,706 reviews)	ratings only (836,006 ratings)
Musical Instruments	5-core (10,261 reviews)	ratings only (500,176 ratings)
Amazon Instant Video	5-core (37,126 reviews)	ratings only (583,933 ratings)

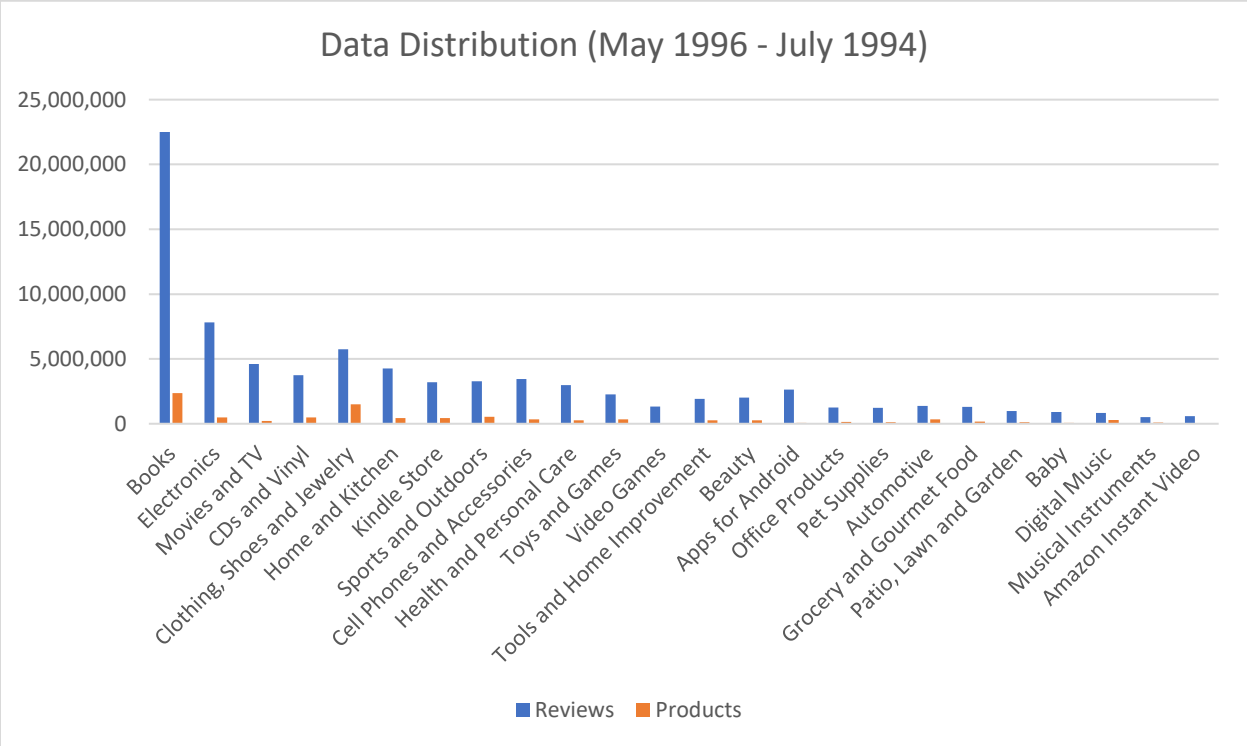
Merging these datasets, resulted in our big “Technology” database, which needed further cleaning. For acquiring a second “safety net” and cleaner data we used the “aggressively deduplicated database”, provided from the repository, which has no duplicates whatsoever and contains 82.83 million of reviews, in contrast with the 142.8 million of the raw review database. Then we “mapped” our “Technology” data on the deduplicated one, to acquire our final duplicated-free dataset. We achieved this by performing an inner-join of the two datasets on the “productID”.

A useful feature that was needed for our future statistical analysis was the brand of each product. The repository provided us this data indirectly, by having metadata sets, which included descriptions, price, sales-rank, brand info, and co-purchasing links. Once again, we merged the relative metadata sets into one bigger dataset, which we then “mapped” on our existing “Technology” database by performing an inner-join on the “productID” and keeping only the relative “Brand” column. Unfortunately not all the products were assigned their brand, meaning

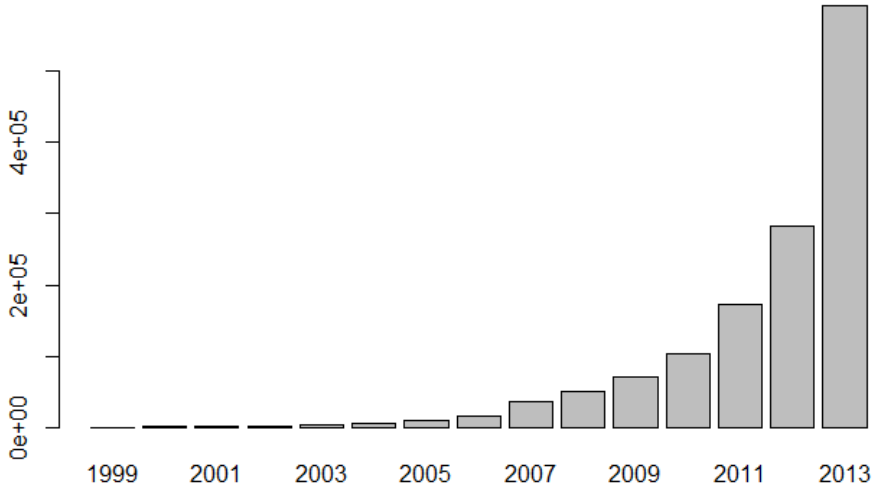
that we had to manually code for the rest of the products and match them with their respective brand. Following the steps described above, we managed to create our final dataset, on which our further analysis will depend on.

It should be noted, that after a meeting with the company, we were advised to initially perform our analysis on a smaller dataset, and then expand our actions on the larger one, so we are currently operating on the Electronics 5-core subset.

The following chart gives the distribution of number of reviews and products available under each category.



Further, under electronics category, the number of reviews available each year are as follows;



Data Description:

In this section, we will discuss the attributes of fields in the datasets. Product database contains the following fields:

Name	Description
Asin	ID of the product
imURL	URL of the product image
Description	Description of the product
Categories	Category the product belongs to
title	The title of the product
Price	Price of the product
Related	Related products (also viewed, also bought, buy together)
salesRank	Sales Rank information
Brand	Brand name

Reviews database contains the following fields:

Name	Description
Asin	ID of the product
ReviewerID	Id of the reviewer
ReviewerName	Name of reviewer
helpful	Helpfulness rating of the review
Reviewtext	Text of the review
Overall	Rating of the product
Summary	Summary of the review
unixReviewTime	Time of the review (unix time)
reviewTime	Time of the review (raw)

As evident, both datasets will be linked by the product key column. It is pertinent to mention here that this key is unique only in the products database as multiple reviews can have single product key in reviews database.

Methodology:

Our first goal is to match Company names to products. We have a list of publicly traded companies. The list consists of 644 electronics companies. (ranging from appliances to semi-conductors, we included all companies that could relate to electronics production).

We have the list of the unique products traded under Electronics category for the sample period. We subset this list to products that have at least 1 review. We have around 63000 distinct products traded in the dataset with at least 1 review.

The Matching Process:

We remove all punctuations from title, description and company names column. In addition, we transmute all columns to lower case letters. At this point these are the all modifications to the data.

1) At first attempt we try to find in which titles we can find the full name of the company:

Example:

the example company name is: **21Vianet Group, Inc.**

Then we remove punctuation and make it lower case: **21vianet group inc**

The first element in title column is: **kelby training dvd mastering blend modes in 21vianet group inc adobe photoshop cs5 by corey barker**

If the company name (all words in order with spaces) is found in the title, then we have a match. In the above example we have a match since the company name appears fully in the title. In case we are missing at least one part of the company name then it will not count as a match. These are restrictive rules which we will attenuate later.

We have 644 companies and each entry in the table represents the number of products that were matched to the company name by the above algorithm:

Total: 8 matches. Very bad algorithm for matching

2) After seeing that looking in titles does not work as we want, we try to search in description column which contains many more words than title column.

Total: 337 matches. A little bit better but still not even close to what we want.

3) Since description column also didn't provide us with more information, we try to modify company names further. We try to remove from company names repetitive and not relevant words. We search company names column for the most repetitive words and remove them from company names. The removed words are:

```
 [1] "inc"           "corporation"   "ltd"           "holdings"      "technologies"
 [6] "international" "limited"       "group"         "systems"       "corp"
[11] "technology"    "incorporated"  "software"      "solutions"     "networks"
[16] "plc"           "holding"       "n.v"           "company"       "communications"
```

For example, the word “inc” appears in 40% of the data. We assume that this will unlikely appear in the title or description string and therefore remove it along with others.

Under this scheme we have the following:

If before we had: **21vianet group inc**

Then now we will have: **21vianet**

After applying the above algorithm, we have some problems. The process trims too harshly some company names:

Example:

American software inc → American

Avid technology → avid

Therefore, we manually change some company names back to the level when they can be uniquely associated with products:

Some more examples;

Aware → aware inc

Cool → cool holdings

Box → box inc

3d →3d systems corporation

Evolving →evolving systems

Extreme →extreme networks

We consider that this still can represent the company in the reasonable way. Let’s see the results:

We try again to search titles for the full match of modified company names:

Total: 22842 matches. This is a considerable improvement! Covers 40% of products data.

4) Now we try the same for description column:

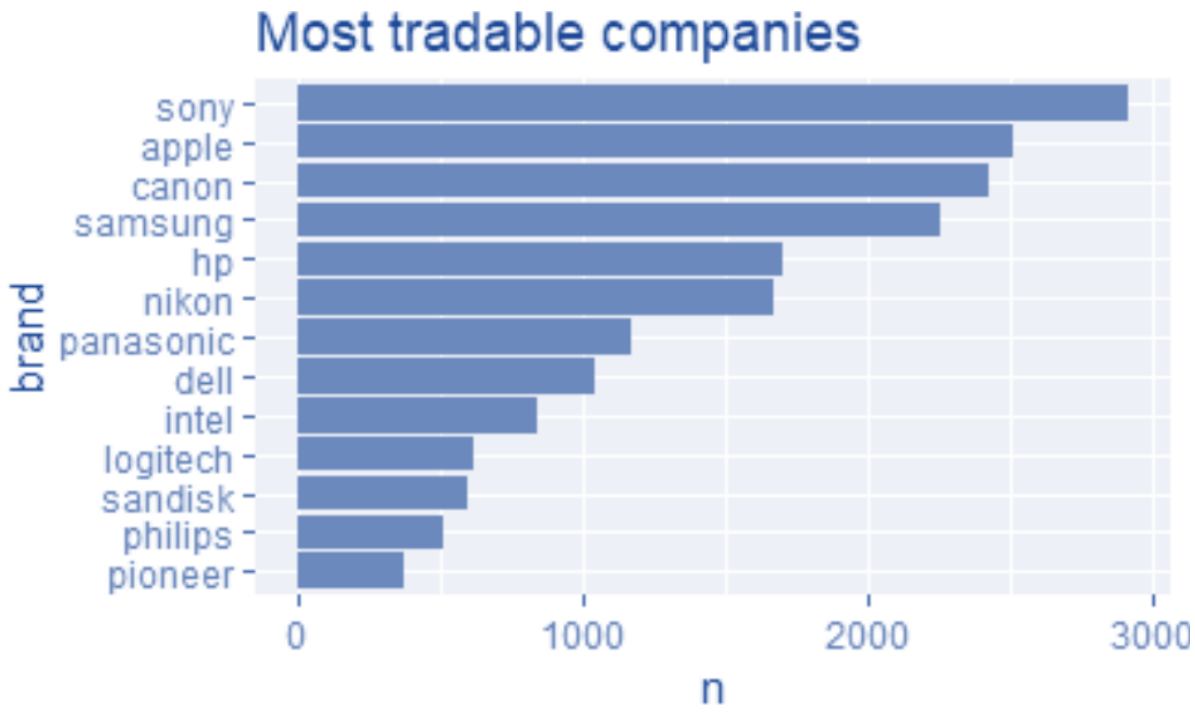
Total: 96707 matches. Cover 180% of the data which is impossible.

The conclusion is that some companies start sharing the same products. For example: if the description contains both “apple” and “sony” then they will be counted to both companies.

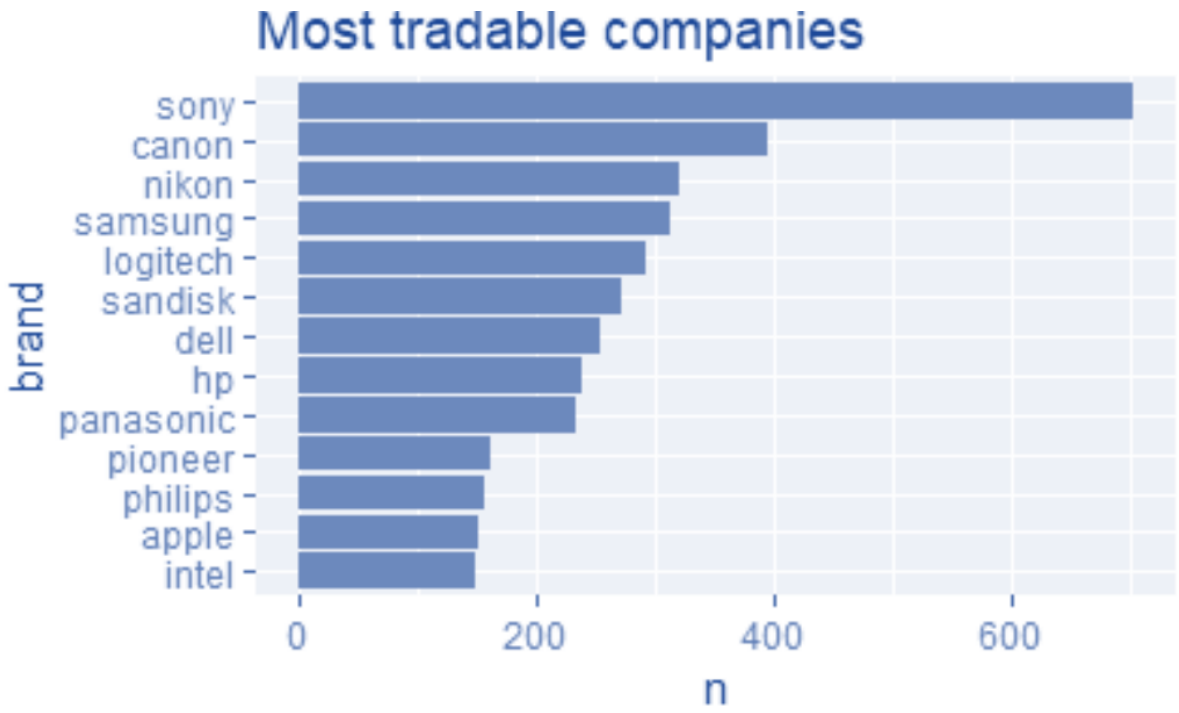
What we did so far? We found how many products with reviews each company has.

What now? We will pick up 13 biggest companies from the list and will concentrate our analysis on them.

Number of products for each company based on Title column search



Number of products for each company based on Brand column search



The difference when I combine brand+title as it turns out is only 18789-18675=114 additional pdocuts added

The accuracy of the model is 92.2%

accuracy=length(unique(tidy_data3$asin_num))/length(tidy_data3$asin_num)

They cover 18789 products out of 22842 products. So, 13 companies cover 18789/22842% of all the data and the rest 100 companies cover remaining percentage of Data. This implies that most products on Amazon are dominated by large companies.

Sentiment Analysis:

We established connection between company names and products.

Next step will be to work with review database and identify sentiment for each review. We do this before merging data sets together. The reason is that to merge all textual data of reviews to company names and products will cumbersome and in order to save time we will;

1) Identify sentiment for each review

2) Merge only the numeric sentiment score for the product and the date when the review was written.

So, we start working with review dataset:

Step1. Remove reviews of the products that weren't match to any of the product that is included in our 13 firms . We are left with around 500 000 reviews.

Step2. Then we work with review text:

A) make it lower case

B) remove punctuations

C) remove special characters and extra spaces

D) remove numbers

E) We don't remove stopwords dictionary that is built in R text mining package because it contains amplifiers, deamplifiers and negations such as very, not and etc:

```
  [1] "i"          "me"         "my"         "myself"     "we"         "our"        "ours"
  [8] "ourselves"  "you"        "your"       "yours"      "yourself"   "yourselves" "he"
 [15] "him"        "his"        "himself"    "she"        "her"        "hers"       "herself"
 [22] "it"         "its"        "itself"     "they"       "them"       "their"      "theirs"
 [29] "themselves" "what"       "which"      "who"        "whom"       "this"       "that"
 [36] "these"      "those"      "am"         "is"         "are"        "was"        "were"
 [43] "be"         "been"       "being"      "have"       "has"        "had"        "having"
 [50] "do"         "does"       "did"        "doing"      "would"      "should"     "could"
 [57] "ought"      "i'm"        "you're"     "he's"       "she's"      "it's"       "we're"
 [64] "they're"    "i've"       "you've"     "we've"      "they've"    "i'd"        "you'd"
 [71] "he'd"       "she'd"      "we'd"       "they'd"     "i'll"       "you'll"     "he'll"
 [78] "she'll"     "we'll"      "they'll"    "isn't"      "aren't"     "wasn't"     "weren't"
 [85] "hasn't"     "haven't"    "hadn't"     "doesn't"    "don't"      "didn't"     "won't"
 [92] "wouldn't"   "shan't"     "shouldn't"  "can't"      "cannot"     "couldn't"   "mustn't"
 [99] "let's"      "that's"     "who's"      "what's"     "here's"     "there's"    "when's"
[106] "where's"    "why's"      "how's"      "a"          "an"         "the"        "and"
[113] "but"        "if"         "or"         "because"    "as"         "until"      "while"
[120] "of"         "at"         "by"         "for"        "with"       "about"      "against"
[127] "between"    "into"       "through"    "during"     "before"     "after"      "above"
[134] "below"      "to"         "from"       "up"         "down"       "in"         "out"
[141] "on"         "off"        "over"       "under"      "again"      "further"    "then"
[148] "once"       "here"       "there"      "when"       "where"      "why"        "how"
[155] "all"        "any"        "both"       "each"       "few"        "more"       "most"
[162] "other"      "some"       "such"       "no"         "nor"        "not"        "only"
[169] "own"        "same"       "so"         "than"       "too"        "very"
```

F) remove empty strings

G) We will not perform stemming because it doesn't make sense for adjectives (we will identify sentiment solely on adjectives at this stage)

Step3. We use bing dataset to identify the sentiment:

Bing snippet: Total of 2005 positive words and 4781 negative words

	word	sentiment
1	2-faces	negative
2	abnormal	negative
3	abolish	negative
4	abominable	negative
5	abominably	negative
6	abominate	negative
7	abomination	negative
8	abort	negative
9	aborted	negative
10	aborts	negative
11	abound	positive
12	abounds	positive
13	abrade	negative
14	abrasive	negative
15	abrupt	negative
16	abruptly	negative

Step4. All the above steps can be neatly performed by R function: polarity.[1] (note: info is taken from RDocumentation website)

From now and on we will describe what additional steps it undertakes:

Polarity function

The equation used by the algorithm to assign value to polarity of each sentence fist utilizes the sentiment dictionary (Hu and Liu, 2004) to tag polarized words. A context cluster (x_i^T) of words is pulled from around this polarized word (default 4 words before and two words after) to be considered as valence shifters. The words in this context cluster are tagged as neutral (x_i^0) negator (x_i^N) amplifier (x_i^α), or de-amplifier (x_i^d). Neutral words hold no value in the equation but do affect word count (**n**). Each polarized word is then weighted **w** based on the weights from the 'polarity.frame' argument and then further weighted by the number and position of the valence shifters directly surrounding the positive or negative word. The researcher may provide a weight **c** to be utilized with amplifiers/de-amplifiers (default is .8; de-amplifier weight is constrained to -1 lower bound). Last, these context cluster (x_i^T) are summed and divided by the

[1] https://www.rdocumentation.org/packages/qdap/versions/2.3.2/topics/polarity

square root of the word count ($\sqrt{\boldsymbol{n}}$) yielding an unbounded polarity score (**δ**) . Note that context clusters containing a comma before the polarized word will only consider words found after the comma.

$$\delta = \frac{x_i^T}{\sqrt{n}}$$

Where:

$$x_i^T = \sum \left((1 + c(x_i^A - x_i^D)) \cdot w(-1)^{\sum x_i^N} \right)$$

$$x_i^A = \sum (w_{neg} \cdot x_i^a)$$

$$x_i^D = \max(x_i^{D'}, -1)$$

$$x_i^{D'} = \sum (-w_{neg} \cdot x_i^a + x_i^d)$$

$$w_{neg} = \left(\sum x_i^N \right) \bmod 2$$

The equation used by the algorithm to assign value to polarity of each sentence fist utilizes the sentiment dictionary to tag polarized words. Each paragraph composed of sentences, is broken into element sentences. Each sentence is broken into an ordered bag of words. Punctuation is removed with the exception of pause punctuations (commas, colons, semicolons) which are considered a word within the sentence. The words in each sentence are searched and compared to a dictionary of polarized words (e.g., Jockers (2017) dictionary found in the lexicon package). Positive and negative words are tagged with a *+1* and *-1* respectively. These will form a polar cluster which is a subset of a sentence.

The polarized context cluster of words is pulled from around the polarized word and defaults to 4 words before and two words after *pw*) to be considered as valence shifters. The words in this polarized context cluster are tagged as neutral, negator, amplifier, or de-amplifier. Neutral words hold no value in the equation but do affect word count. Each polarized word is then weighted (*w*) based on the weights from the polarity_dt argument and then further weighted by the function and number of the valence shifters directly surrounding the positive or negative word. Pause locations (punctuation that denotes a pause including commas, colons, and semicolons)

are indexed and considered in calculating the upper and lower bounds in the polarized context cluster. This is because these marks indicate a change in thought and words prior are not necessarily connected with words after these punctuation marks. The lower and upper bound of polarized context cluster are also constrained

The core value in the cluster, the polarized word is acted upon by valence shifters. Amplifiers (intensifiers) increase the polarity by 1.8 (.8 is the default weight). Amplifiers become de-amplifiers if the context cluster contains an odd number of negators. De-amplifiers (downtoners) work to decrease the polarity. Negation acts on amplifiers/de-amplifiers as discussed but also flip the sign of the polarized word. Negation is determined by raising -1 to the power of the number of negators +2. Simply, this is a result of a belief that two negatives equal a positive, 3 negatives a negative and so on.

The adversative conjunctions (i.e., 'but', 'however', and 'although') also weight the context cluster. A adversative conjunction before the polarized word up-weights the cluster by a specific amount given from a formula. An adversative conjunction after the polarized word down-weights the cluster by a specific amount also given from a specific formula.The number of occurrences before and after the polarized word are multiplied by 1 and -1 respectively and then summed within context cluster. It is this value that is multiplied by the weight and added to 1. This corresponds to the belief that an adversative conjunction makes the next clause of greater values while lowering the value placed on the prior clause.

This is how the polarity function provides score in the last column:

	asin	text	score	time	polarity
1	0528881469	well can say ive unit truck four days now prior garmin t n...	3	2010-09-09	0.51784389
2	0528881469	going write long review even thought unit deserves one i...	2	2010-11-24	-0.06851887
3	0528881469	im professional otr truck driver bought tnd truck stop ho...	1	2010-11-25	-0.06950480
4	0528881469	ive mine year heres got tries route non truck routes tellin...	1	2011-09-29	-0.39605902
5	0528881469	got gps husband otr road trucker impressed shipping tim...	5	2013-06-02	1.38564065
6	8862936826	read many reviews folio cases ipad will find best ever revi...	5	2010-11-29	1.20021366
7	8862936826	ive waiting cover everything thought moleskine quality s...	1	2010-11-30	0.17614097
8	8862936826	im starting review changing stars like stars dont like first ...	2	2010-11-30	1.03091175
9	8862936826	product appeared high quality well made also feature tur...	4	2011-12-17	0.40824829
10	8862936826	owned month hoping use meeting situations ipad typing...	3	2013-02-16	0.57370973

It correlates with score column for 34%

Step5.

Now it's time to merge firms, products and sentiment in one data table:

	company_name	time_vec	sentiment_vec
1	sony	c("1999-07-23", "1999-10-13", "1999-10-20", "1999-12-09...	c(-0.252907426928101, 0.154303349962092, 1.0153369346...
2	canon	c("1999-12-31", "2000-05-02", "2000-06-08", "2000-06-17...	c(-0.133333333333333, 0.91706052144883, 1.46686059323...
3	nikon	c("1999-10-26", "1999-11-03", "1999-11-15", "1999-12-02...	c(-0.158943882847805, 0.126491106406735, -0.474341649...
4	samsung	c("2000-12-31", "2001-05-04", "2001-09-05", "2001-12-04...	c(0.784398431204706, 1.29875199971232, 1.05065722146...
5	logitech	c("2000-05-23", "2000-06-24", "2000-08-13", "2000-08-14...	c(0.832050294337844, 0, 1.18594462200587, 0.998687663...
6	sandisk	c("2000-06-16", "2000-06-17", "2000-08-06", "2000-09-07...	c(1.10858717169259, 0.351123441588392, 0.11396057645...
7	dell	c("2002-12-25", "2003-01-04", "2003-01-13", "2003-03-01...	c(0.539359889970594, 0.92689738158054, -0.14744195615...
8	hp	c("2000-06-02", "2000-08-06", "2000-08-20", "2000-09-06...	c(0.0333333333333333, 0.514614016722141, 0.991647658...
9	panasonic	c("1999-07-08", "1999-09-02", "1999-11-23", "1999-12-05...	c(0.366508333068916, 0, 0.859246812473437, 1.05531237...
10	philips	c("1999-11-17", "1999-11-27", "1999-12-01", "1999-12-08...	c(0.7184212081071, 0.567698757436222, 1.502637680879...
11	apple	c("2001-09-14", "2001-09-22", "2001-09-26", "2001-10-19...	c(0.612372435695795, -0.171498585142509, 0.7483314773...
12	pioneer	c("2000-05-16", "2000-08-02", "2000-11-15", "2000-12-19...	c(0.896179253777926, 1.24151489048701, 0.83445538495...
13	intel	c("2000-04-24", "2000-04-28", "2000-04-30", "2000-05-02...	c(0.758011398851084, 1.95544352835329, 1.21065445546...

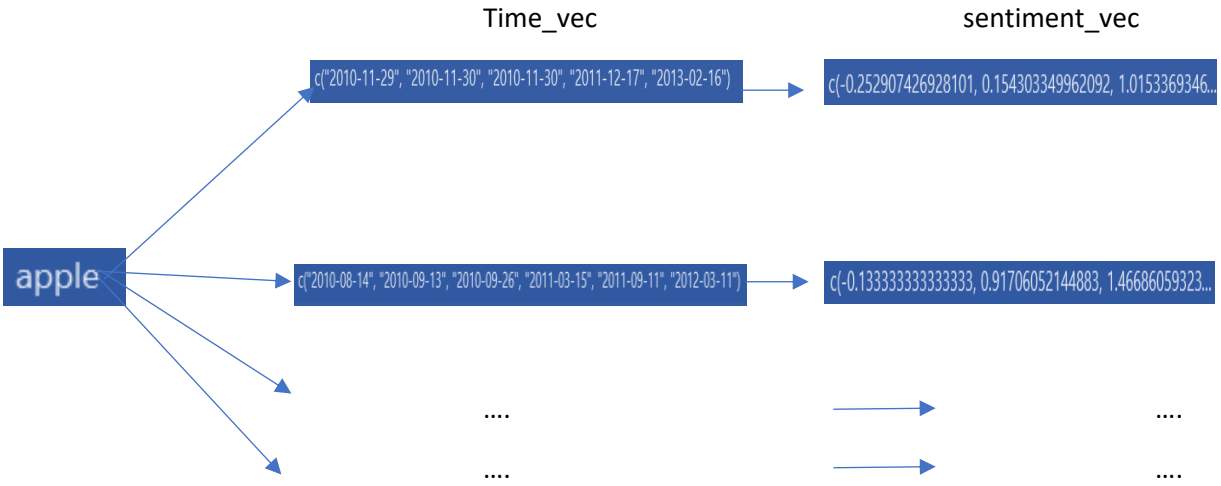
There are 13 rows where each row represents a company.

Time_vec and sentiment_vec represent the following:

Time_vec represents a particular day and Sentiment_vec represents simple sum or weighted average of sentiment scores of all products for a particular company over a specified date.

Let's demonstrate it on the example how it was constructed:

Each arrow represents a product of Apple. For each product we have certain amount of reviews. Time vec shows the dates of these reviews and sentiment vec shows the corresponding sentiment score for each date.



Then there are two ways to go. There are simple and weighted method for combining vectors:

Simple Method:

We sum up sentiment score for each date for each product

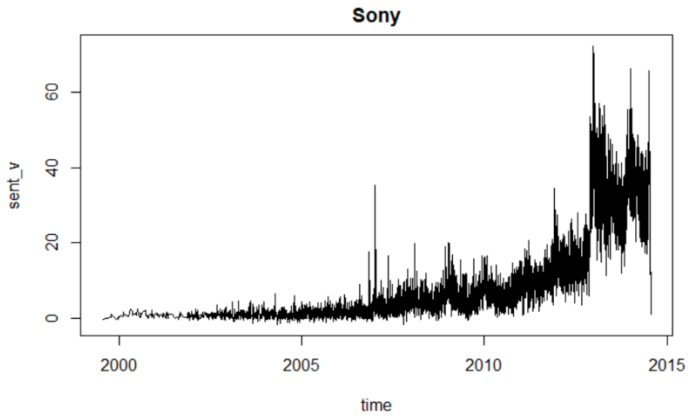
Weighted Method:

If on a given day for product A we have 4 reviews with sentiment score -0.5 1 1.5 1 then we find simple weighted average of these numbers 0.75. This will represent sentiment score for product A on a given day.

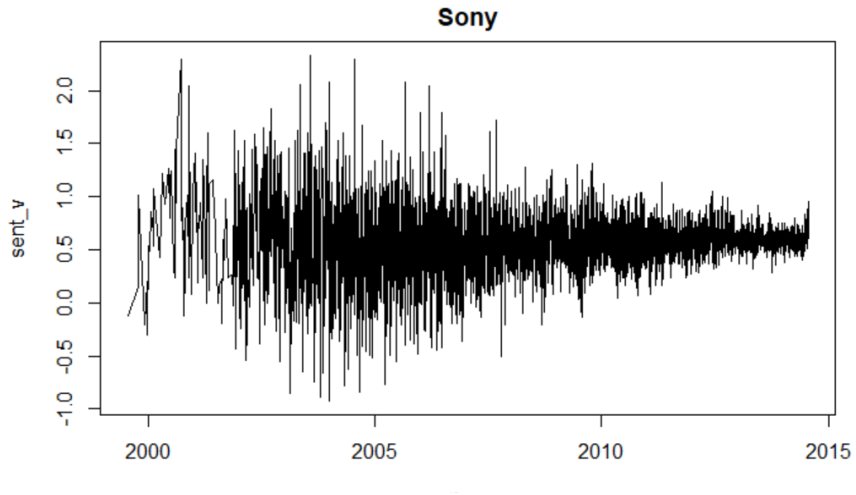
Since we have multiple products and we want to find one number to represent sentiment for the company on a given day we take value weighted average of sentiments across products where weights are based on number of reviews. Products with more comments should receive more weight.

If we have three products A,B,C and A has 2 reviews while B,C have only one review each then the weighting scheme will be 0.5,0.25,0.25.

If we plot sentiment over time of **Simple method** then we would see something like this (example based on company Sony):



If we do the same with **Weighted method** then the picture looks like we are dealing with returns:



Fundamentals explained by Sentiment:

For each company we construct quarterly sentiment index. Each quarter consists of three months and we have, in most cases, sentiment score for each day. We equally weigh each day within 3 months period and thus obtain one sentiment score for each quarter.

After calculating the sentiment score, we calculated the Pearson Correlation of total sentiment score over the quarter with next quarter's fundamentals. The choice of fundaments was arbitrary, however, we tried to incorporate all items that may be affected by the change in sentiment scores for the big companies mentioned in the table below. For our analysis, median and median/standard deviation may give an indication of which variable are mostly likely affected by the sentiment of reviews on average. It may be considered that the analysis so far

has only been carried out on the big companies and results may change if, for example, stocks from small or medium cap are taken into account.

As seen from the table below, Cost of Goods Sold (COGS) has the highest correlation over dispersion compared to its counterparts. The absolute median correlation of 36.2% also seems significant over a period from 2010 to 2014 June (18 Quarters). Revenue also has a decent median correlation of 30.3% but its not as significant as COGS. We believe that as overall sentiment score for product increases, the demand for that item increases which puts supply pressure on the companies, especially if most of their sales are dependent on amazon. Also, in order to meet the expenditures from short term production requirements, current liabilities may increases as well. Further, increasing supply also pushes inventory, and possibly PPE up, which increases total assets as well. One of the characteristics of electronics category is that people often find it convenient to purchase the products online and that could be one of the reason why we are able to see some decent correlation. Because overall reviews of products may also indicate its overall perception in the market and therefore this sentiment score may give us an idea if a product will be successful in future.

	sony	canon	logitech	sandisk	hp	panasonic	apple	intel	max	min	median	median/sd
Cost of Goods Sold	0.125	0.372	0.495	0.026	0.468	0.482	-0.021	0.353	0.495	-0.021	0.362	1.827
Total Assets	0.294	0.387	0.348	-0.061	0.387	0.598	-0.293	0.357	0.598	-0.293	0.353	1.318
Current Liabilities	0.430	0.427	0.346	-0.209	0.147	0.615	-0.245	0.527	0.615	-0.245	0.386	1.264
Revenue	0.100	0.284	0.471	-0.243	0.412	0.474	-0.019	0.247	0.474	-0.243	0.265	1.113
Liabilities	0.284	0.395	0.321	-0.151	0.277	0.663	-0.423	0.485	0.663	-0.423	0.303	0.920
Common Equity	-0.029	0.349	0.288	0.018	0.260	0.463	-0.178	0.082	0.463	-0.178	0.171	0.841
Current Assets	0.159	0.341	0.157	-0.060	0.115	0.530	-0.242	0.211	0.530	-0.242	0.158	0.720
Cash	0.087	0.340	-0.237	0.259	-0.361	0.356	-0.241	0.210	0.356	-0.361	0.148	0.548
Cash/ST Investments	0.013	0.300	-0.237	0.127	-0.355	0.372	-0.294	-0.015	0.372	-0.355	-0.001	-0.003
Retained Earnings	-0.187	-0.067	0.358	0.059	0.175	0.420	-0.148	-0.137	0.420	-0.187	-0.004	-0.018
EPS 12MM	-0.353	0.071	-0.075	-0.399	0.026	-0.198	0.480	0.120	0.480	-0.399	-0.024	-0.092
Common Shares	-0.418	0.618	0.203	0.425	0.287	-0.349	-0.916	-0.513	0.618	-0.916	-0.073	-0.144
EPS Operations	-0.017	-0.085	0.208	-0.264	-0.103	-0.174	0.491	-0.205	0.491	-0.264	-0.094	-0.400
Net Income	-0.018	0.025	0.072	-0.181	-0.151	-0.189	0.108	-0.328	0.108	-0.328	-0.084	-0.594

Current Results and Future Plan:

We have concentrated on Electronics database from Amazon. After matching companies to products on the database we have selected 13 companies that have the biggest number of products traded. These 13 companies cover 90% of the data which means that the rest 100 companies matched to database share only 10% of all products. This tells us that if we want to analyze smaller companies in the future then we will have problems due to sparsity of data.

After selecting 13 companies we further narrow down the list to 8 companies because the rest companies do not belong to US universe (example: Samsung, Nikon, Pioneer).

We perform sentiment analysis for these 8 companies based on R function “polarity”. The function uses Bing dictionary of positive and negative words. It incorporates the impact of amplifiers (very, pretty and etc.), deamplifiers and negators(not, never and etc). It also adjusts score for the number of words in the review. After obtaining polarity score for each review we group data where every company has date and the sentiment on this date. The sentiment on a particular date comprises of two weighting schemes. We equally weigh the sentiment score for each product and then we value weigh obtained sentiment score from the previous step across products where weight depends on the number of reviews each product has (the more reviews product has the more weight it will receive).

We want to see how sentiment score is related to different fundamentals of the company. We group sentiment scores by quarters (where each day sentiment score receives equal weight). Then we compare the created factor to fundamentals of the companies. We presented the correlation matrix in which the following patterns were revealed: COGS and Total Assets have highest median correlation with previous quarter’s sentiment score when standardized by the dispersion among companies. Four of the fundaments have correlation higher than 30% which is decent in terms of predicting the fundamentals so far. (Refer to table above for more details)

Moving forward, we are planning to see how sentiment score is correlated with stock returns of the companies on different time intervals.

References:

Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering
R. He, J. McAuley
WWW, 2016

Image-based recommendations on styles and substitutes
J. McAuley, C. Targett, J. Shi, A. van den Hengel
SIGIR, 2015

More Than Words: Quantifying Language to Measure Firms' Fundamentals
Tetlock,
et. Al, 2008

Lazy Prices
Cohen,
et. al 2015

Textual Analysis Slides
Lars, Lochstoer
2019

R Documentation

Rinker, Tyler

2012

Appendix: (Code Appended Below)

Text Based Signals

Rustem Shinkaruk

September 28, 2019

```
library(data.table)
library(jsonlite)
library(tidytext)
library(dplyr)
library(tidyverse)
library(tm)
library(stringr)
library(readxl)
library(DT)
library(textdata)
library(class)
library(gmodels)
library(ggplot2)
rev=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\data.csv"))#reviews
products=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\products.csv"))#products
comp=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\companylist1.csv"))#company list

products=products[products$asin %in% unique(rev$asin),]#choose products only for which we have reviews
```

```
matched_brands=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\brands matched.csv"))
comp=matched_brands$brand[1:20]
comp=comp[-2]
comp=comp[c(1:9,15,16,18,19)]
```

Here I match only by titles

```
titles<-removePunctuation(products$title)#create vector of titles
titles <- tolower(titles)
complementary=!is.na(titles)
titles=titles[complementary]#remove na titles
data3=comp

#create vector with products asin numbers
prodAsin=products$asin
prodAsin=prodAsin[complementary]

#create datatable to fill in later
df=data.table(data3)
setnames(df,c("name"))

#create vector to hold number of products traded for each company
num_products<- vector("numeric",length(data3))

#
index_place=list()
asin_list=list()

for(i in 1:length(data3)){
```

```
  num_products[i]=sum(str_detect(titles,pattern=data3[i]))
  dd=which(str_detect(titles,pattern=data3[i])==TRUE)
  if(length(dd)!=0){
    index_place[[i]]=dd
    asin_list[[i]]=prodAsin[dd]
  }else{
    index_place[[i]]=0
    asin_list[[i]]=0
  }
}

df[,num_products:=num_products,]
df[,index_place:=index_place,]
df[,asin_list:=asin_list,]

main=df[num_products!=0]
main=main[,.(name,asin_list)]
#vectorize asin_list column, make it one string separated by space
main[,asin:=0,]
for(i in 1:length(main$name)){
  main$asin[i]=paste(unlist(main$asin_list[i]),collapse=" ")
}

main[,asin_list:=NULL,]
setnames(main,c("name","asin_vec"))

#now use unnest_token to make the data tidy

tidy_data <- main%>%
  unnest_tokens(asin_num,asin_vec)

tidy_data$asin_num=toupper(tidy_data$asin_num)
```

Now I show the number of products for each company matched only by titles

```
x=tidy_data%>%
  count(name)%>%
  arrange(desc(n))


x=as.data.table(x)

words_count <- x%>%
  mutate(brand=fct_reorder(name,n))
ggplot(words_count,aes(x=brand,y=n))+geom_col()+coord_flip()+ggtitle("Most tradable companies")
```

Now Try to do based only on brand column (not titles column)

```
brand <- removePunctuation(products$brand)
brand <- tolower(brand)
temp_dt=data.table(brand)
complementary=!is.na(temp_dt$brand)
temp_dt <- na.omit(temp_dt)

titles_brand=temp_dt$brand
```

```r
#create vector with products asin numbers
prodAsin=products$asin
prodAsin=prodAsin[complementary]

#create datatable to fill in later
df=data.table(data3)
setnames(df,c("name"))

#create vector to hold number of products traded for each company
num_products<- vector("numeric",length(data3))

#
index_place=list()
asin_list=list()

for(i in 1:length(data3)){
  num_products[i]=sum(str_detect(titles_brand,pattern=fixed(data3[i])))
  dd=which(str_detect(titles_brand,pattern=data3[i])==TRUE)
  if(length(dd)!=0){
    index_place[[i]]=dd
    asin_list[[i]]=prodAsin[dd]
  }else{
    index_place[[i]]=0
    asin_list[[i]]=0
  }
  #print(i)
}
#================================================================================
#================================================================================
df[,num_products:=num_products,]
df[,index_place:=index_place,]
df[,asin_list:=asin_list,]

main=df[num_products!=0]
main=main[,.(name,asin_list)]
#vectorize asin_list column, make it one string separated by space

for(i in 1:length(main$name)){
  main$asin[i]=paste(unlist(main$asin_list[i]),collapse=" ")
}

main[,asin_list:=NULL,]
setnames(main,c("name","asin_vec"))

#now use unnest_token to make the data tidy

tidy_data2 <- main%>%
  unnest_tokens(asin_num,asin_vec)

tidy_data2$asin_num=toupper(tidy_data2$asin_num)
```

```
x=tidy_data2%>%
  count(name)%>%
  arrange(desc(n))


x=as.data.table(x)

words_count <- x%>%
  mutate(brand=fct_reorder(name,n))
ggplot(words_count,aes(x=brand,y=n))+geom_col()+coord_flip()+ggtitle("Most tradable companies")
```

this is not necessary. It shows how many unque brands are there on amazon database and shows a number of products for each brand

```
y=brand
y=na.omit(y)
y=data.table(y)
setnames(y,c("name"))
y=y%>%
  count(name)%>%
  arrange(desc(n))


y=as.data.table(y)
```

Now let's combine brands and titles results into one data table. The difference as it turns out is only 18789-18675=114 additional pdocuts added

```
# brand <- removePunctuation(products$brand)
# brand <- tolower(brand)
# titles<-removePunctuation(products$title)#create vector of titles
# titles <- tolower(titles)
#
# temp_dt=data.table(titles,brand)
#
# #temp_dt=temp_dt[-which(is.na(titles) & is.na(brand)),]
#
# temp_dt[,title_brand:=ifelse((!is.na(titles) & !is.na(brand)),paste(titles,brand),ifelse(!is.na(title
#
#
# titles_brand=temp_dt$title_brand
# complementary=!is.na(titles_brand)
# titles_brand=na.omit(titles_brand)
# #create vector with products asin numbers
# prodAsin=products$asin
# prodAsin=prodAsin[complementary]
#
# #create datatable to fill in later
# df=data.table(data3)
# setnames(df,c("name"))
#
# #create vector to hold number of products traded for each company
# num_products<- vector("numeric",length(data3))
#
# #
```

```
# index_place=list()
# asin_list=list()
#
# for(i in 1:length(data3)){
#   num_products[i]=sum(str_detect(titles_brand,pattern=fixed(data3[i])))
#   dd=which(str_detect(titles_brand,pattern=data3[i])==TRUE)
#   if(length(dd)!=0){
#     index_place[[i]]=dd
#     asin_list[[i]]=prodAsin[dd]
#   }else{
#     index_place[[i]]=0
#     asin_list[[i]]=0
#   }
#   #print(i)
# }
# #=============================================================================
# #=============================================================================
# df[,num_products:=num_products,]
# df[,index_place:=index_place,]
# df[,asin_list:=asin_list,]
#
#
# main=main[,.(name,asin_list)]
# #vectorize asin_list column, make it one string separated by space
# main[,asin:=0,]
# for(i in 1:length(main$name)){
#   main$asin[i]=paste(unlist(main$asin_list[i]),collapse=" ")
# }
#
# main[,asin_list:=NULL,]
# setnames(main,c("name","asin_vec"))
#
# #now use unnest_token to make the data tidy
#
# tidy_data3 <- main%>%
#   unnest_tokens(asin_num,asin_vec)
#
# tidy_data3$asin_num=toupper(tidy_data3$asin_num)


#write.csv(tidy_data3,"C:\\Users\\Rustem\\Desktop\\afp 2\\tidy_data3.csv", row.names = FALSE)
tidy_data3=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\tidy_data3.csv"))
```

Cleaning reviews for sentiment analysis

```
# cleaned_reviews=rev[rev$asin %in% unique(tidy_data3$asin_num),]
# cleaned_reviews[,reviewerID:=NULL,]
# cleaned_reviews[,reviewerName:=NULL,]
# cleaned_reviews[,unixReviewTime:=NULL,]
# cleaned_reviews$reviewTime=as.Date(cleaned_reviews$reviewTime,format="%m %d, %Y")
#
#
# setnames(cleaned_reviews,c("asin","text","score","summary","time"))
# setorderv(cleaned_reviews,c("asin","time"))
```

```
# #============================================================
# #firsly let's remove all digits and punctuations from our text and make all lower letters
# #============================================================
# cleaned_reviews$text=tolower(cleaned_reviews$text)
# cleaned_reviews$text=removePunctuation(cleaned_reviews$text)
# cleaned_reviews$text=removeNumbers(cleaned_reviews$text)
# cleaned_reviews$text=removeWords(cleaned_reviews$text,stopwords("english"))
# cleaned_reviews=cleaned_reviews[-which(cleaned_reviews$text==""),]
# cleaned_reviews$text=stripWhitespace(cleaned_reviews$text)
# cleaned_reviews[,summary:=NULL]
# #====================================================================
# #WE WILL NOT PERFORM STEMMING SINCE IT DOES NOT MAKE SENSE IN THIS CASE
# #====================================================================
# library(textdata)
# a=get_sentiments("bing")
# #write.csv(cleaned_reviews,"C:\\Users\\Rustem\\Desktop\\afp 2\\cleaned_reviews3.csv", row.names = FAL
cleaned_reviews=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\cleaned_reviews3.csv"))
```

Sentiment Analysis: (polarity approach)

```
#
# install.packages("qdap")
# library(qdapRegex)
# library(qdapDictionaries)
# install.packages("rJava")
# library(rJava)
# #Sys.setenv(JAVA_HOME="C:\\Program Files\\Java\\jre1.8.0_221") # for 32-bit version
# library(qdap)


#write.csv(pol_df,"C:\\Users\\Rustem\\Desktop\\afp 2\\pol_df2.csv", row.names = FALSE)
pol_df=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\pol_df2.csv"))
```

check correlation with score

```
yy=data.table(pol_df$score,pol_df$polarity)
yy=na.omit(yy)
cor(yy$V1,yy$V2)
```

Accuracy of the model

```
accuracy=length(unique(tidy_data3$asin_num))/length(tidy_data3$asin_num)

accuracy
```

Group polarity score and time by firm

```
tidy_data3[,time_vec:=0,]
tidy_data3[,sentiment_vec:=0,]

for(i in 1:length(tidy_data3$asin_num)){
  aa=pol_df[asin==tidy_data3$asin_num[i],]
  l1=list(aa$time,aa$polarity)
  tidy_data3$time_vec[i]=l1[1]
  tidy_data3$sentiment_vec[i]=l1[2]
  if(i%%250==0){
```

```
    print(i)
  }
}
```

```
final=data.table()
final[,company_name:=unique(tidy_data3$name),]
final[,time_vec:=0,]
final[,sentiment_vec:=0,]
for(i in 1:length(final$company_name)){
  aa=tidy_data3[name==final$company_name[i],]
  master_dt=data.table()
  for(j in 1:length(which(tidy_data3$name==final$company_name[i]))){
    dt1=data.table(unlist(aa$time_vec[j]),unlist(aa$sentiment_vec[j]))
    master_dt=rbind(master_dt,dt1)
  }
  x=master_dt%>%
    group_by(V1)%>%
    summarize(sum(V2))
  l1=list(x$V1,x$`sum(V2)`)
  final$time_vec[i]=l1[1]
  final$sentiment_vec[i]=l1[2]

  print(i)
}
```

```
time=as.Date(unlist(final$time_vec[1]),format="%Y-%m-%d")
sent_v=unlist(final$sentiment_vec[1])
plot(time,sent_v,type="l",main="Sony")
```

```
final_complex=data.table()
final_complex[,company_name:=unique(tidy_data3$name),]
final_complex[,time_vec:=0,]
final_complex[,sentiment_vec:=0,]
for(i in 1:length(final_complex$company_name)){
  aa=tidy_data3[name==final_complex$company_name[i],]
  master_dt=data.table()
  for(j in 1:length(which(tidy_data3$name==final_complex$company_name[i]))){
    dt1=data.table(rep(aa$asin_num[j],length(unlist(aa$time_vec[j]))),unlist(aa$time_vec[j]),unlist(aa$s
    master_dt=rbind(master_dt,dt1)
  }
  master_dt[,weight:=1/(.N),by=.(V1,V2)]
  master_dt[,weighted_sent:=V3*weight]

  master_dt[,num_reviews:=(.N),by=.(V1,V2)]
  master_dt[,total_reviews:=(.N),by=.(V2)]
  master_dt[,day_weight:=num_reviews/total_reviews]
  x=master_dt[,list(col1=sum(weighted_sent),col2=day_weight),by=.(V1,V2)]
  x=unique(x)


  setnames(x,c("asin","time","lambda","weights"))
```

```
  x[,weighted_sent_day:=weights*lambda]
  x=x[,sum(weighted_sent_day),by=c("time")]
  setorderv(x,c("time"))

  l1=list(x$time,x$V1)
  final_complex$time_vec[i]=l1[1]
  final_complex$sentiment_vec[i]=l1[2]

  print(i)
}
```

```
time=as.Date(unlist(final_complex$time_vec[1]),format="%Y-%m-%d")
sent_v=unlist(final_complex$sentiment_vec[1])
plot(time,sent_v,type="l",main="Sony")
```

```
fund=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\fundamentals.csv"))
sony=fund[189:226,]
sony$datadate=as.Date(as.character(sony$datadate),format="%Y%m%d")
```

For sony we have data for each data through 4,5 years

```
dt_sony=data.table(time=as.Date(unlist(final_complex$time_vec[1]),format="%Y-%m-%d"),sent_v=unlist(final_complex$sentiment_vec[1]))

dt_sony[,year:=year(time)]
dt_sony[,month:=month(time)]
dt_sony[,quarter:=ifelse(month<=3,1,ifelse(month<=6,2,ifelse(month<=9,3,ifelse(month<=12,4,4))))]
dt_sony[,period:=paste(year,"Q",quarter,sep="")]
dt_sony[,year:=NULL]
dt_sony[,month:=NULL]
dt_sony[,quarter:=NULL]
dt_sony=dt_sony[time>="2010-01-01" & time<="2014-06-01",]

dt_sony=na.omit(dt_sony)
sup=dt_sony%>%
  group_by(period)%>%
  summarise(total=sum(sent_v))

sup=as.data.table(sup)
sup[,normalized_sent:=(total-mean(total))/(sd(total))]


sony=sony[datadate>="2010-01-01" & datadate<="2014-06-30",]

cor(sup$normalized_sent,sony[,c(15:28)])
```

```
master_cor=data.table()
for(i in 1:length(final_complex$company_name)){
  time=as.Date(unlist(final_complex$time_vec[i]),format="%Y-%m-%d")
  sentiment=unlist(final_complex$sentiment_vec[i])
  dt=data.table(time,sentiment)[time>="2010-01-01" & time<="2014-06-01",]
  dt[,year:=year(time)]
  dt[,month:=month(time)]
  dt[,quarter:=ifelse(month<=3,1,ifelse(month<=6,2,ifelse(month<=9,3,ifelse(month<=12,4,4))))]
  dt[,period:=paste(year,"Q",quarter,sep="")]
  dt[,year:=NULL]
```

```
  dt[,month:=NULL]
  dt[,quarter:=NULL]

  dt=na.omit(dt)
  sup=dt%>%
    group_by(period)%>%
    summarise(total=sum(sentiment))
  sup=as.data.table(sup)
  sup[,normalized_sent:=(total-mean(total))/(sd(total))]



  firm=fund[which(conm==final_complex$company_name[i]),]
  firm$datadate=as.Date(as.character(firm$datadate),format="%Y%m%d")
  firm=firm[datadate>="2010-01-01" & datadate<="2014-06-30",]


  if(length(firm$gvkey)==0){
    next
  }

  if(final_complex$company_name[i]=="dell"){
    next
  }
  y=cor(sup$normalized_sent,firm[,c(15:28)])
  master_cor=rbind(master_cor,y)
 # eval(parse(text=paste("dt_",final_complex$company_name[i],"=dt",sep="")))

}



master_cor=cbind(data.table(final_complex$company_name[c(1,2,5,6,8,9,11,13)]),master_cor)
#write.csv(master_cor,"C:\\Users\\Rustem\\Desktop\\afp 2\\master_cor.csv", row.names = FALSE)
master_cor=as.data.table(fread("C:\\Users\\Rustem\\Desktop\\afp 2\\master_cor.csv"))
```

```
vec=data.table("max")
for(i in 2:15){
  vec=cbind(vec,max(master_cor[[i]]))
}
setnames(vec,colnames(master_cor))
vec1=data.table("min")
for(i in 2:15){
  vec1=cbind(vec1,min(master_cor[[i]]))
}
setnames(vec1,colnames(master_cor))
vec2=data.table("meadian")
for(i in 2:15){
  vec2=cbind(vec2,median(master_cor[[i]]))
}
setnames(vec2,colnames(master_cor))

master_cor=rbind(master_cor,vec,vec1,vec2)
```