Analysis of Stock Symbol Co-occurrences in Financial Articles

Gregory Kramida (gkramida@cs.umd.edu)

Introduction

Stock market prices are influenced by a wide variety of factors. Undoubtedly, market news articles written by financial analysts are very influential: many small investors do not have the luxury of professionals managing their portfolios, and do not have the time to become experts themselves, hence, they rely on reading the free and readily available stock reviews and financial news by accredited authors to understand which stocks to buy, which – to sell, and when. However, just by looking at headlines or single articles themselves, it is difficult to assess the overall value of this enormous media collection: are we getting the full picture, and how complete is the coverage? Do the reviewers tend to be biased? Are there any strong patterns, and what are they?

This project addresses these and other issues by looking at co-occurrences of stock symbols ("tickers") in financial articles. It also addresses certain stock trends themselves. How are companies related to each other? Which companies are regarded as adversaries, and which as friends? Which companies across different sectors and industries tend to be mentioned together most often, and why? Perhaps visualizing this network can shed some light.

Scraping the Data

Initially, the only dataset readily available from www.nasdaq.com was one of U.S. publicly-traded companies[1] and mutual funds. It spanned the NYSE, NASDAQ, and AMEX exchanges, and contained ticker symbols, full company names, market capitalization, sector, and industry. Naturally, since mutual funds are composed of many stocks, only the tickers and names were available for these.

The actual dataset mentioned – a thorough count of ticker co-occurrences in articles – did not exist before this project was started. To actually build the dataset, I developed a web spider that would crawl through the Internet and scrape the co-occurrences. This was accomplished with the aid of the scrapy library for the python programming language. What also greatly simplified the task was that the financial news article collection on the NASDAQ website amassed articles from multiple diverse and respectable sources, e.g. Zachs Investment Research, Street Authority, Seeking Alpha, MT Newswires, and Benzinga, to name a few, and presented them in a uniform (with regard to html) format.

What slightly complicated the task was that tickers for the same companies can officially change. Ticker change information was available at the NASDAQ website, but not in cohesive format – spread among many pages in html lists. A separate web spider was made to collect this information, and 915 ticker changes were obtained, first one dating to 2009.

The Dataset

The primary web spider went through all available articles, numbering over 50K, dating from around April 2009 up until now, and logged the ticker co-occurrences within the body of each article. With the article date available, ticker changes were taken into account, and the outdated tickers entered the dataset in their newest forms. A total of 412K+ co-occurrences were logged.

---

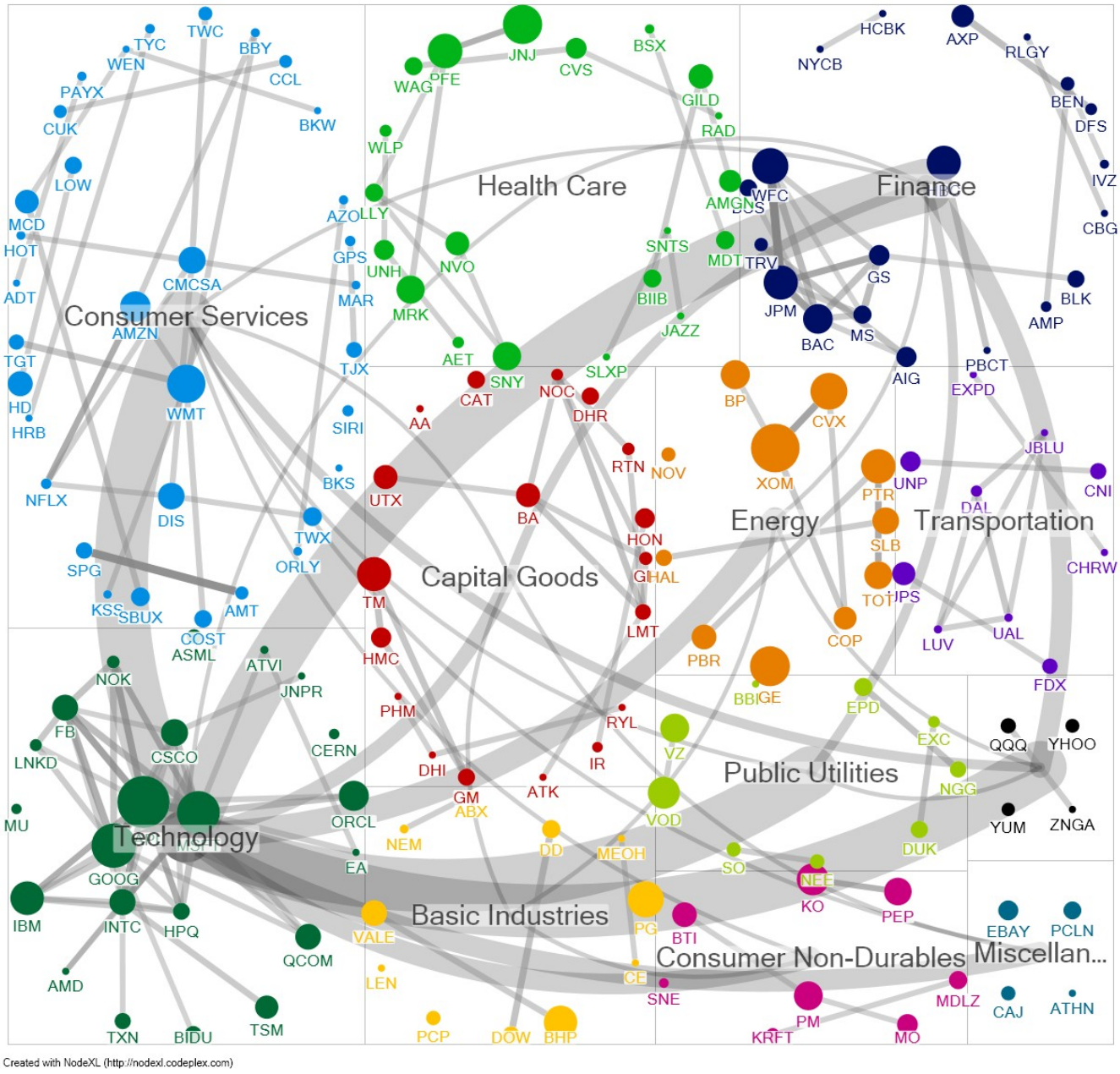
1 Due to the time and resource constraints, I chose to limit the scope to U.S. stock exchanges

Some noteworthy statistics[2] about the dataset were immediately revealing: only 1% of all the mutual funds ever got mentioned together with other tickers. This is quite dazzling: if the articles were indeed targeting small investors, with mutual funds being a popular commodity among small inverters, one would think they would have better coverage.

The dataset was imported into PostgreSQL and an aggregate edge set was compiled, shedding the information about individual articles but compiling the total co-occurrence count for every pair of tickers, arriving at 200K+ aggregate edges.

2 Note: the statistics reflect only those articles that mentioned more than one ticker symbol, or about 20%. While this is not the entire dataset, it is still statistically-representative of the whole.

Everybody Needs Tech



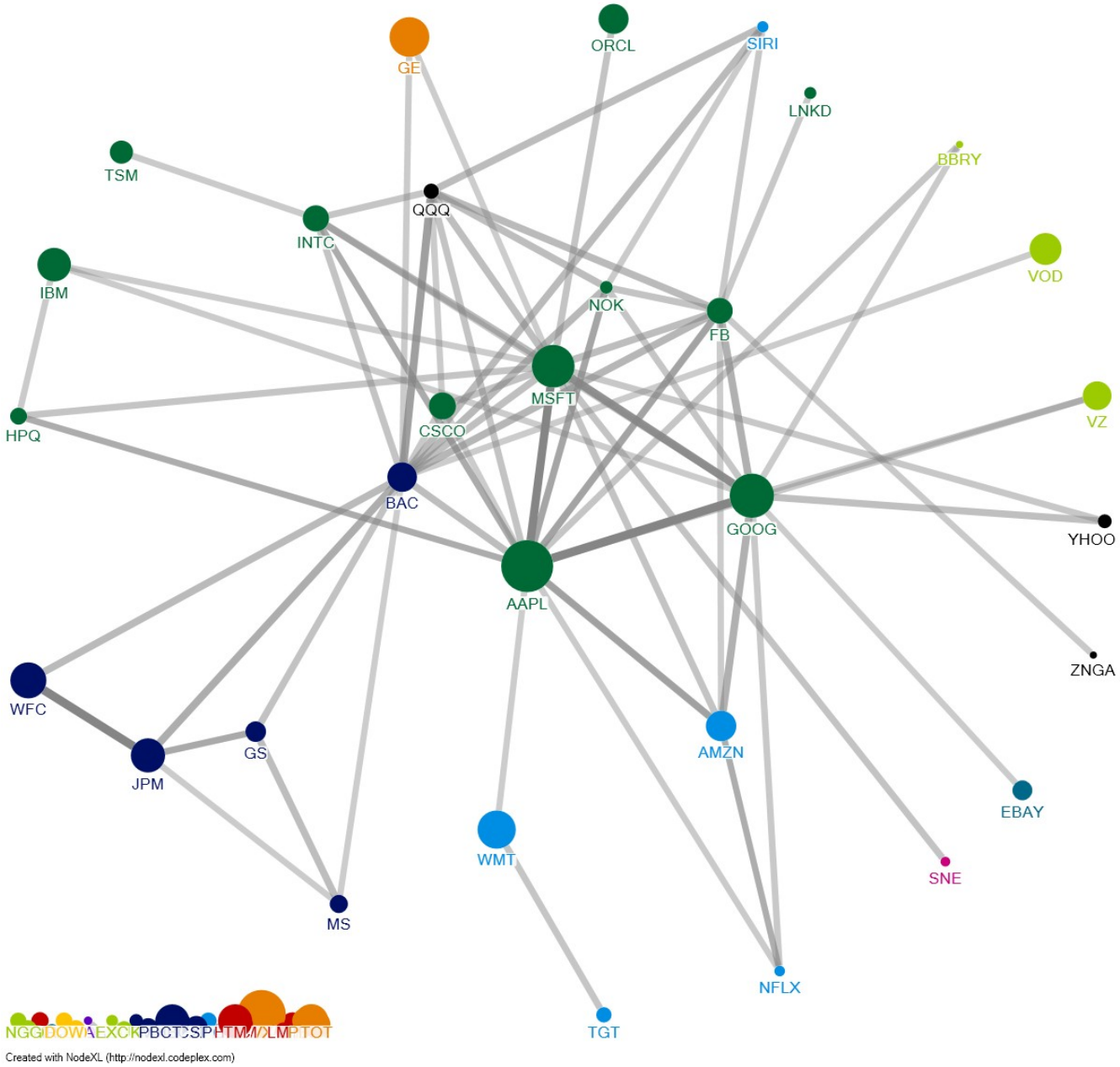
Created with NodeXL (http://nodexl.codeplex.com)

This NodeXL rendering shows the graph limited to only the strongest edges, for tickers that were mentioned together in over 50 articles. Edge opacity and width both represent this number, node size represents market capitalization, while color represents sector. As the graph suggests, technology does not only have the most connections leading to other sectors, but also has the strongest connections. Indeed, group metrics reveal that 26% of the connections between the sectors terminate in technology, followed by 10% for both Finance and Consumer Services. The leading players by betweenness centrality are Apple (AAPL), Bank of America (BAC), and Microsoft (MSFT). The most secluded sector is transportation, where the strongest inter-sector link is 42-times co-mention of Delta Air Lines (DAL) and Boeing (BA), too weak to be displayed here.

What this suggests is that technology is more relevant to other companies everywhere, and also prompts the question whether “Technology” is a valid sector at all: if everything uses technology,

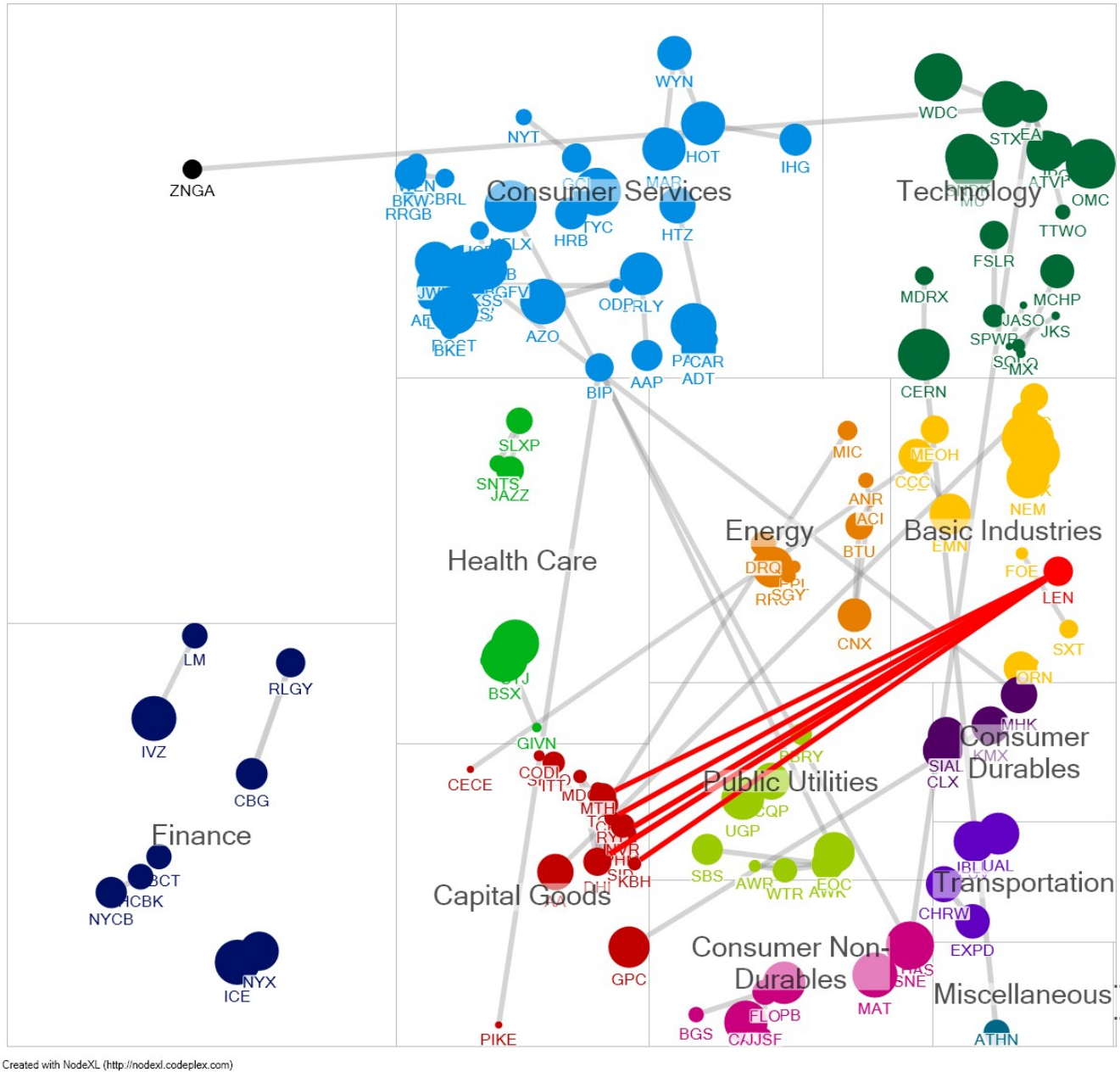
perhaps it shouldn't appear as a separate sector, and the companies within should be redistributed to other sectors.

Next to the Giants



This cluster graph provides more insight into which smaller companies get mentioned more often. It holds only the 100x co-occurrence and stronger edges. It is no surprise that AAPL and MSFT share the strongest 745-article edge in the graph: these are giant companies rivaling in the same sector and industry. Much more interesting is that companies that are comparatively small get mentioned more often due to their overall popularity or their ties with the “giants”: United Financial (PBCT) is linked to HSBC Holdings, the tiny LinkedIn (LNKD) and Zynga (ZNGA) stand alongside Facebook (FB), Netflix (NFLX) enjoys mention next to Apple, Google, and Amazon, while Nokia (NOK) benefits from close ties with Microsoft. Having much room for growth and such popularity, these small companies may be regarded good prospective investments.

Uncovering Classification Errors



Created with NodeXL (http://nodexl.codeplex.com)

Following the same trend, I decided to look at smaller (but well-connected) fish – companies with market capitalization of less than $200M. Although the picture was less rewarding from the financial standpoint, several problems were uncovered with the sector classification NASDAQ used for the companies. For instance Lennar Corporation, a large house construction company, got pushed into “Basic Industries”, with a ton of connections leading to other construction companies appearing in “Capital Goods”. All appear in the “Homebuilding” industry, which asks whether the sector label is correct. Cerner Corporation (CERN) and Athena Health (ATHN) compete within the health insurance industry, but got assigned to “Technology” and “Miscellaneous” respectively.

These findings effectively demonstrate that group visualizations may be used to find classification errors.

Visualizations & NodeXL Critique

NodeXL was used to visualize the data. The main benefit that NodeXL provided was using opacity and width edge properties, as well as laying out companies grouped by industry in their own containers, was critical to make sense of the graph, even with only the strongest connections left in place.

In a perfect world, individual co-occurrence edges would be inserted to allow NodeXL to compute graph metrics, such as degree and various centralities, more accurately. It quickly became obvious that even the 200K+ *aggregate* edge set was unfeasible to work with due to performance limitations of the software[3]. I had to filter out edges with single co-occurrences using SQL, which plunged the number of aggregate edges down to 56477 and vertices down to 3910.

Thus the main problem is obvious: for large networks, NodeXL fails to provide an overview of the whole data. Following Ben Shniederman's “Overview, zoom & filter, details-on-demand” mantra, it is obvious that zooming and filtering cannot be done effectively without an appropriate overview. I suggest the developers to break with the limitations Excel imposes (clearly unsuitable for exploring large networks), replacing the back-end with a real RDBMS. OpenCL should be used to parallelize computation and direct hardware acceleration should be used for graph display.

Both Excel built-in filters and NodeXL dynamic filters had to be used to explore the dataset right away. However, NodeXL doesn't integrate well with the Excel filters for nodes. When I tried to limit the dataset to a particular subset of nodes (e.g. based on market capitalization), it was still displaying edges to nodes that were “filtered out,” rendering the task impossible without repeated preprocessing and re-import using SQL. This is something that seems like it could be easily fixed and would make a world of difference. Dynamic filtering still doesn't solve the problem, because it constrains the layout with the “invisible” nodes that are filtered out. Filtering edges proved to be more useful.

3 It took over 15 minutes to lay out and render the whole data just once, in the plainest format available.

What may be more useful to the developers is this error message that popped up while the program was computing graph metrics: