Mohammed Momin

Trading Algorithm from Sentiment Analysis

Abstract

For this project, I have evaluated two different approaches to acquiring sentiment analysis from 10-K filings to create a trading algorithm. For the first approach, I used the SentiWordNet lexical resource to find a positive and negative sentiment score from 10-K filings of a set of stocks within a given time-period. For the second approach, I used the Loughran-McDonald Master Dictionary to find sentiment for words in a 10-K. I only used positive and negative sentiments of the dictionary. The algorithm initiated a buy signal when the positive sentiment score is higher than the negative sentiment score and a sell signal vice versa.

Introduction

Publicly traded companies in the U.S. are required to file a host of documents with the Securities and Exchange Commission (SEC) with one of them being a 10-K report. The 10-K report is a full description of the company's financial activity during a given fiscal year and a full rundown of risks, legalities, liabilities, corporate agreements, operations, and market performance. It is comprised of a total of 14 sections and in a language as specified by SEC Rule 421(d), which advises firms to comply with plain English principles, including short sentences, active voice, and no multiple negatives [2]. Information from this report gives a comprehensive overview of the overall health of this company but reading each document individually would take very long. For this project, I have used information retrieval to acquire sentiment analysis from 10-K filings, without reading them. That sentiment was then used to create a trading strategy for a set of stocks.

Background

10-K documents are public record on the SEC's website; however, they are not as easy to parse as just any url. The SEC forbids any automated tool except those that are approved to parse through 10-K urls simply by using a python script. This required me to use an api to acquire individual sections of each 10-K document and combine them together to create one document. Each 10-K url is found by a query I used to search for 10-K documents by a company's unique Central Index Key (CIK).

Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Various tools in information retrieval can help us identify sentiment from text documents. In this project, I investigate two approaches: one for SentiWordNet and one for the Loughran-McDonald Master Dictionary. SentiWordNet is a public lexical resource used for research purposes. In SentiWordNet, each synset is associated with a score for positive, negative, and neutral sentiment [1]. The Loughran-McDonald Master Dictionary is a list of words that are specifically used for 10-Ks and associates them to a series of categories: positive, negative, uncertainty, litigious, strong modal, weak modal, constraining, and complexity. The word lists are linked to 10-K filing returns, trading volume, return volatility,

fraud, material weakness, and unexpected earnings [3]. In this project, we compare these two approaches in terms of acquiring sentiment from 10-K documents to create a trading algorithm for a given set of stocks.

Approach

The first step is to select a list of stocks, which I selected as AMZN, IBM, BA, GE, and DIS for this project. For each company, we will need to find the unique CIK (Central Index Key) to retrieve their 10-Ks. A list of links for 10-K documents can be found using the url 'https://www.sec.gov/cgi-bin/browse-edgar?action=getcompany &CIK={}&type={}&start={}&count={}&owner=exclude&output=atom' where CIK is the the central index key, the type is 10-K, and the start and count will determine how many document links are queried. For this project we started at 0 and had a count of 60. Once the 10-K filing links for each stock are found, they must be stored in a dictionary for easy access. That dictionary will be used along with the SEC_api to get the text of each 10-K document and store it in another dictionary. Once the data is acquired and stored, it must be preprocessed which includes converting all text to lowercase, lemmatization of words, and removing stop words. From here, the project separates into two parts:

1. SentiWordNet

The preprocessed data is used to create a bag of words from 10-K documents using SentiWordNet. For each word, SentiWordNet gives a score for 3 categories: positive, negative, and neutral. This project takes the highest score between negative and positive and associates that sentiment to the specific word. This will create a sentiment word list which will associate each word to its sentiment. This sentiment word list is then used to generate sentiment term frequency–inverse document frequency (TFIDF) from the 10-K documents creating vectors for each sentiment. To compare sentiments to each filing, this project uses cosine similarity to calculate the cosine of the angle between the sentiment curves and each 10-K document. The cosine similarity curves are then used to generate buy and sell signals. If the positive cosine similarity is greater than the negative cosine similarity, then the algorithm generates a buy signal for the given date of the SEC filing, otherwise it generates a sell signal.

2. Loughran-McDonald

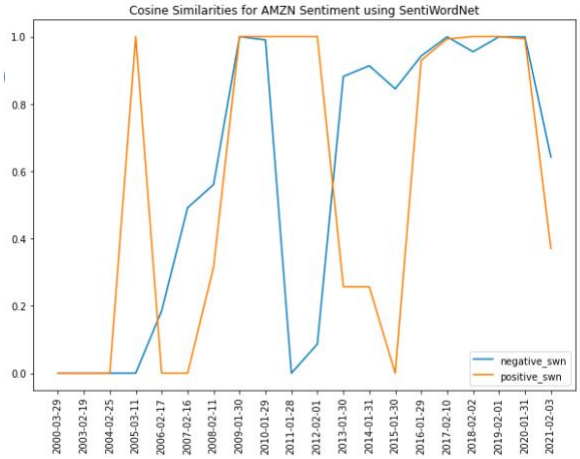
This is very similar to the first approach but instead of using SentiWordNet to calculate sentiment scores, this approach uses the Loughran-McDonald Master Dictionary to associate each word to a given sentiment from the Loughran-McDonald dictionary. Since this dictionary had many more sentiments than SentiWordNet, I narrowed it down to just using its positive and negative sentiments to have a better way of comparing to SentiWordNet. The same way as the first approach, a sentiment word list which is the dictionary itself will be used to generate TFIDF and cosine similarity between the sentiment vectors and 10-K filings vectors. If the positive cosine similarity is greater than

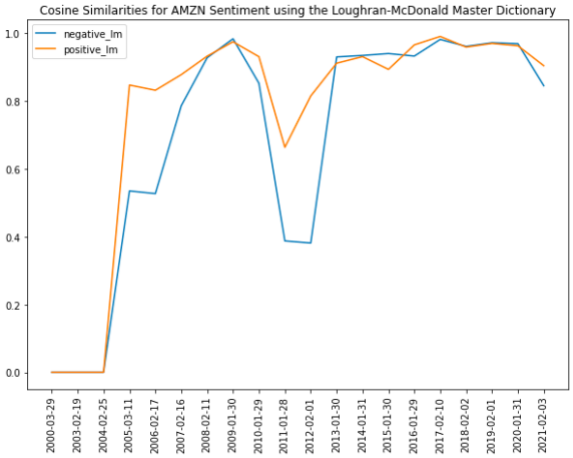
the negative cosine similarity, then the algorithm generates a buy signal for the given date of the SEC filing, otherwise it generates a sell signal.

To backtest the trading signals, 'actual' signals were created where the signal would be buy when the price of the stock at the time of next years 10-K filing is higher than the year being evaluated and sell if next years stock price is lower than the year being evaluated. The 'actual' signals were then compared with each algorithms' signals to determine trading algorithm accuracy.

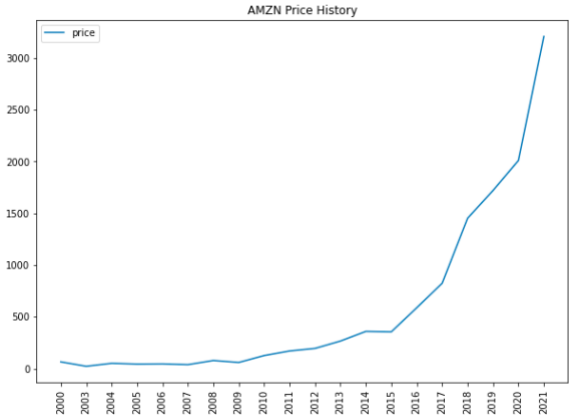
Results

1. Amazon

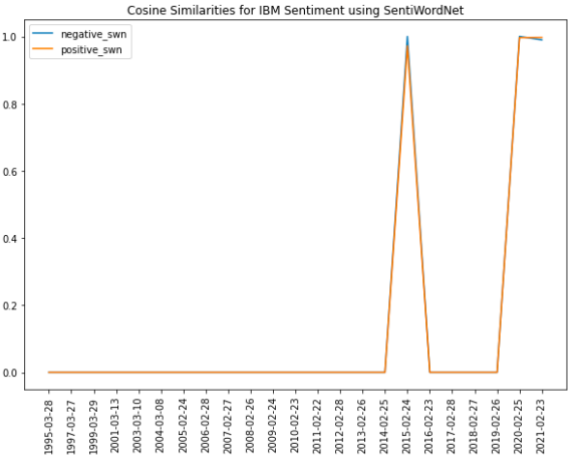


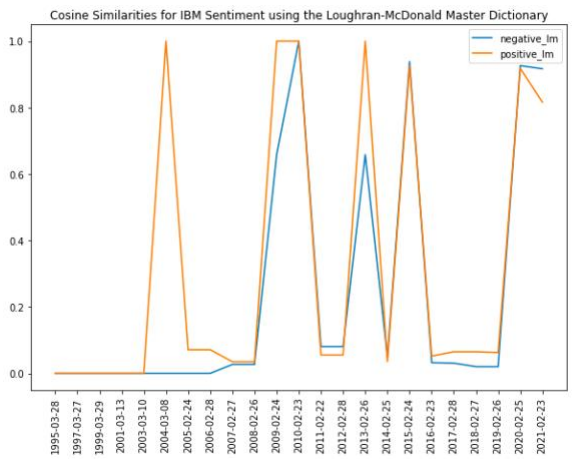


negative_lm	positive_lm	negative_swn	positive_swn	Date	Price	lm_prediction	swn_prediction	actual
0	0	0	0	2000	64.5625	SELL	SELL	SELL
0	0	0	0	2003	21.85	SELL	SELL	BUY
0	0	0	0	2004	50.400002	SELL	SELL	SELL
0.534186277	0.846214285	0	1	2005	43.220001	BUY	BUY	BUY
0.526354291	0.83086543	0.185222848	0	2006	44.82	BUY	SELL	SELL
0.785260215	0.87692785	0.491386769	0	2007	37.669998	BUY	SELL	BUY
0.92671472	0.931350319	0.56054042	0.315164051	2008	77.699997	BUY	SELL	SELL
0.982557828	0.973697368	0.999729693	1	2009	58.82	SELL	BUY	BUY
0.851439336	0.929774096	0.989949494	1	2010	125.410004	BUY	BUY	BUY
0.386957348	0.66267584	0	1	2011	169.639999	BUY	BUY	BUY
0.380736518	0.814049898	0.086629616	1	2012	194.440002	BUY	BUY	BUY
0.929063514	0.91059689	0.881343419	0.256763736	2013	265.5	SELL	SELL	BUY
0.933426085	0.929890003	0.913222849	0.256763736	2014	358.690002	SELL	SELL	SELL
0.939041322	0.892414601	0.844555833	0	2015	354.529999	SELL	SELL	BUY
0.931558533	0.964532732	0.942056138	0.928954546	2016	587	BUY	SELL	BUY
0.980628752	0.989446446	0.999273758	0.992862948	2017	823.47998	BUY	SELL	BUY
0.960185556	0.958020773	0.954765626	1	2018	1450.89002	SELL	BUY	BUY
0.970912603	0.968837017	0.999178162	1	2019	1718.72998	SELL	BUY	BUY
0.968070218	0.961965639	0.998921118	0.992862948	2020	2008.71997	SELL	SELL	BUY
0.844221182	0.903159461	0.641991068	0.370193803	2021	3206.19995	BUY	SELL	BUY



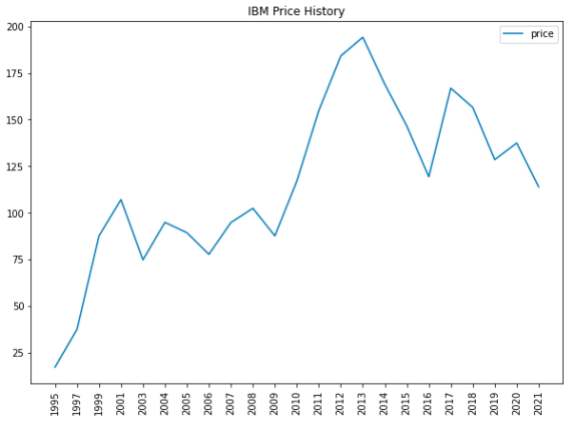
2. IBM



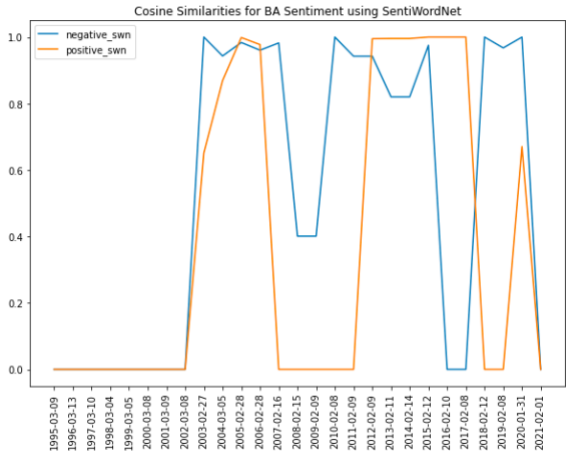


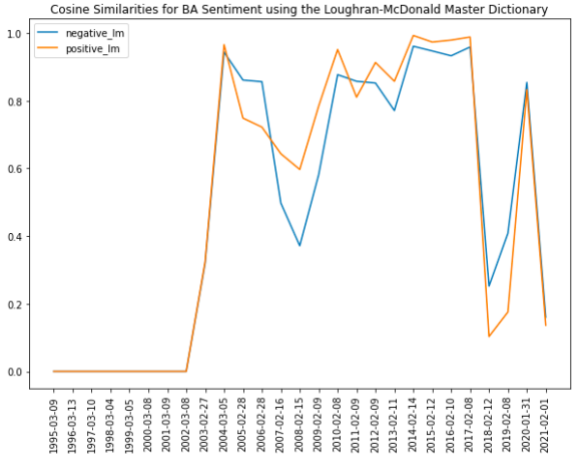
Mohammed Momin

negative_lm	positive_lm	negative_swn	positive_swn	Date	Price	lm_prediction	swn_prediction	actual
0	0	0	0	1995	17.238289	SELL	SELL	BUY
0	0	0	0	1997	37.494026	SELL	SELL	BUY
0	0	0	0	1999	87.595604	SELL	SELL	BUY
0	0	0	0	2001	107.07457	SELL	SELL	SELL
0	0	0	0	2003	74.760994	SELL	SELL	BUY
0	1	0	0	2004	94.866158	BUY	SELL	SELL
0	0.07095907	0	0	2005	89.311661	BUY	SELL	SELL
0	0.07095907	0	0	2006	77.724663	BUY	SELL	BUY
0.026748418	0.03491931	0	0	2007	94.789673	BUY	SELL	BUY
0.026748418	0.03491931	0	0	2008	102.39962	BUY	SELL	SELL
0.658272945	1	0	0	2009	87.619499	BUY	SELL	BUY
1	1	0	0	2010	117.007645	SELL	SELL	BUY
0.080581729	0.055198884	0	0	2011	154.875717	SELL	SELL	BUY
0.080581729	0.055198884	0	0	2012	184.13002	SELL	SELL	BUY
0.658272945	1	0	0	2013	194.139572	BUY	SELL	SELL
0.048953141	0.03533907	0	0	2014	168.910141	SELL	SELL	SELL
0.938446684	0.925329521	1	0.972335974	2015	146.567871	SELL	SELL	SELL
0.032083428	0.051956436	0	0	2016	119.302101	BUY	SELL	BUY
0.030670942	0.064929799	0	0	2017	166.845123	BUY	SELL	SELL
0.020189851	0.064929799	0	0	2018	156.500961	BUY	SELL	SELL
0.020090236	0.06269647	0	0	2019	128.508606	BUY	SELL	BUY
0.926114332	0.918001248	1	0.996545758	2020	137.40918	SELL	SELL	SELL
0.91666829	0.8161515	0.989949494	0.997054486	2021	113.871895	SELL	BUY	SELL

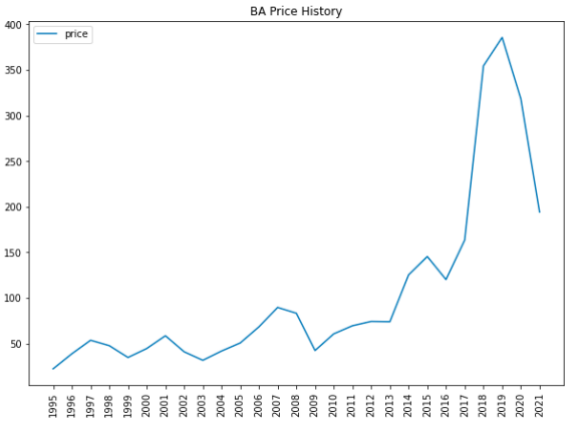


3. Boeing



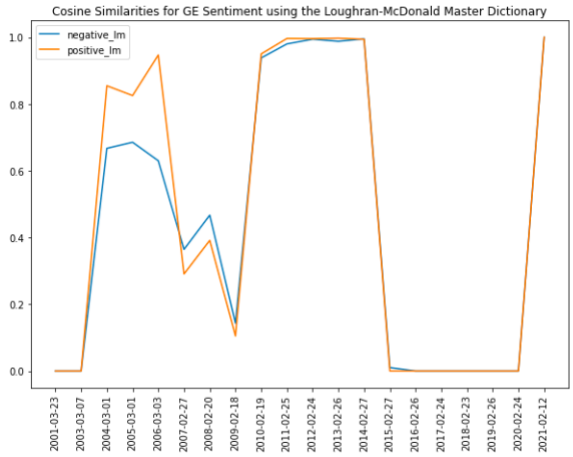


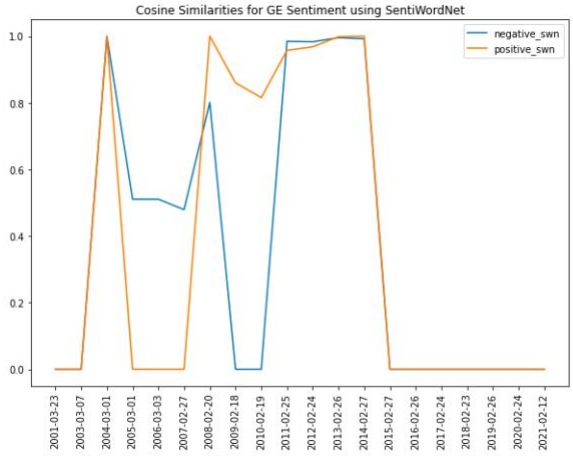
negative_lm	positive_lm	negative_swn	positive_swn	Date	Price	lm_prediction	swn_prediction	actual
0	0	0	0	1995	22.25	SELL	SELL	BUY
0	0	0	0	1996	38.75	SELL	SELL	BUY
0	0	0	0	1997	53.5625	SELL	SELL	SELL
0	0	0	0	1998	47.625	SELL	SELL	SELL
0	0	0	0	1999	34.6875	SELL	SELL	BUY
0	0	0	0	2000	44.5	SELL	SELL	BUY
0	0	0	0	2001	58.5	SELL	SELL	SELL
0	0	0	0	2002	40.950001	SELL	SELL	SELL
0.320073106	0.324001818	1	0.651243526	2003	31.59	BUY	SELL	BUY
0.943685096	0.965713596	0.943106775	0.86886069	2004	41.75	BUY	SELL	BUY
0.861031855	0.74855495	0.983633747	0.998569013	2005	50.599998	SELL	BUY	BUY
0.856287277	0.722023906	0.960515085	0.977337434	2006	68.309998	SELL	BUY	BUY
0.497424277	0.64333505	0.982308282	0	2007	89.559998	BUY	SELL	SELL
0.371281747	0.596858521	0.40089479	0	2008	83.18	BUY	SELL	SELL
0.581667911	0.784345421	0.40089479	0	2009	42.310001	BUY	SELL	BUY
0.87703152	0.951411886	1	0	2010	60.599998	BUY	SELL	BUY
0.857479913	0.810484564	0.942314974	0	2011	69.480003	SELL	SELL	BUY
0.852287903	0.913128354	0.942314974	0.994977403	2012	74.18	BUY	BUY	SELL
0.771004509	0.857511678	0.820160417	0.995664799	2013	73.870003	BUY	BUY	BUY
0.961072503	0.992854553	0.820160417	0.995664799	2014	125.260002	BUY	BUY	BUY
0.947113251	0.973368509	0.975210578	1	2015	145.369995	BUY	BUY	SELL
0.932790667	0.979216631	0	1	2016	120.129997	BUY	BUY	BUY
0.958754359	0.988243726	0	1	2017	163.419998	BUY	BUY	BUY
0.251980638	0.102991443	1	0	2018	354.369995	SELL	SELL	BUY
0.408446845	0.175600474	0.967394234	0	2019	385.619995	SELL	SELL	SELL
0.85408375	0.832931277	0.999793825	0.670275733	2020	318.269989	SELL	SELL	SELL
0.160102878	0.136456671	0	0	2021	194.190002	SELL	SELL	SELL



Mohammed Momin

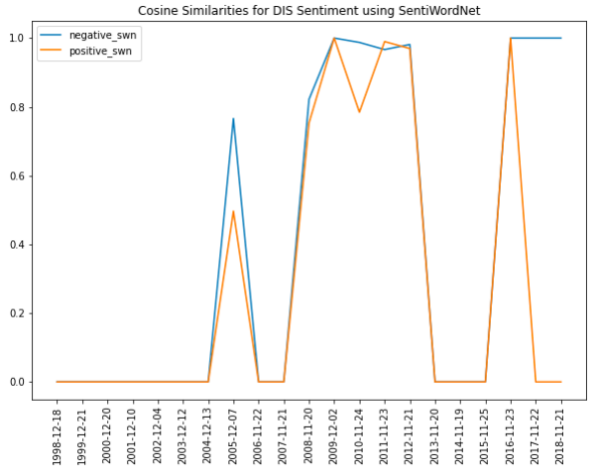
4. General Electric

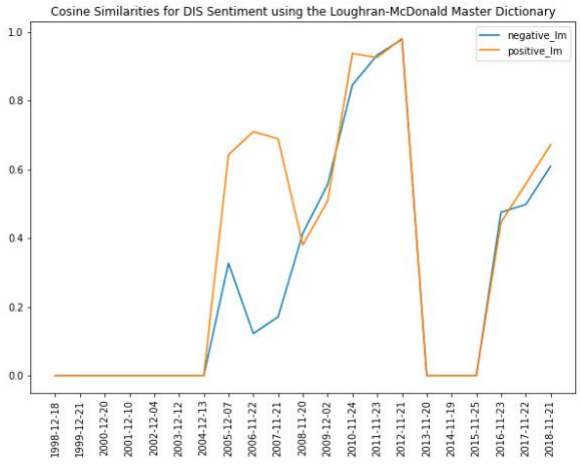




negative_lm	positive_lm	negative_swn	positive_swn	Date	Price	lm_prediction	swn_prediction	actual
0	0	0	0	2001	363.942322	SELL	SELL	SELL
0	0	0	0	2003	197.307693	SELL	SELL	BUY
0.667002592	0.854566965	1	1	2004	244.615387	BUY	SELL	BUY
0.68523324	0.825171627	0.510583792	0	2005	276.923065	BUY	SELL	SELL
0.629826053	0.946732845	0.510583792	0	2006	272.846161	BUY	SELL	BUY
0.364361337	0.290694867	0.479330286	0	2007	288.923065	SELL	SELL	SELL
0.466940746	0.391344094	0.801119457	1	2008	270.538452	SELL	BUY	SELL
0.142992711	0.104871856	0	0.859744536	2009	123.07692	SELL	BUY	BUY
0.93802554	0.950065321	0	0.815435199	2010	127.692307	BUY	BUY	BUY
0.979900718	0.99623772	0.984480488	0.957343942	2011	141.769226	BUY	SELL	BUY
0.994052855	0.995749918	0.983010619	0.967942212	2012	143.461533	BUY	SELL	BUY
0.987737033	0.997080797	0.995420422	0.998176444	2013	162.538467	BUY	BUY	BUY
0.995189375	0.993946251	0.992075021	1	2014	207.384613	SELL	BUY	SELL
0.010187674	0	0	0	2015	184.846161	SELL	SELL	BUY
0	0	0	0	2016	218.846161	SELL	SELL	BUY
0	0	0	0	2017	243.153839	SELL	SELL	SELL
0	0	0	0	2018	142.615387	SELL	SELL	SELL
0	0	0	0	2019	68.769234	SELL	SELL	BUY
0	0	0	0	2020	93.360001	SELL	SELL	SELL
1	1	0	0	2021	90.720001	SELL	SELL	BUY

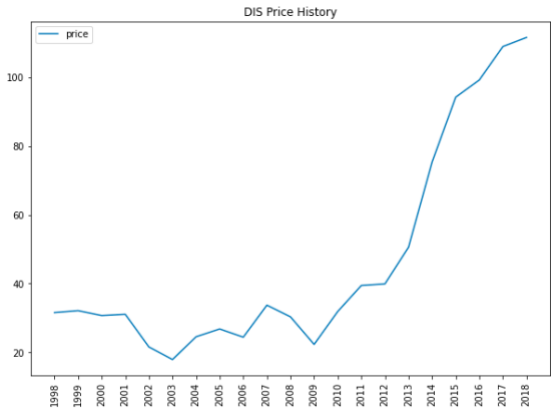
5. Disney

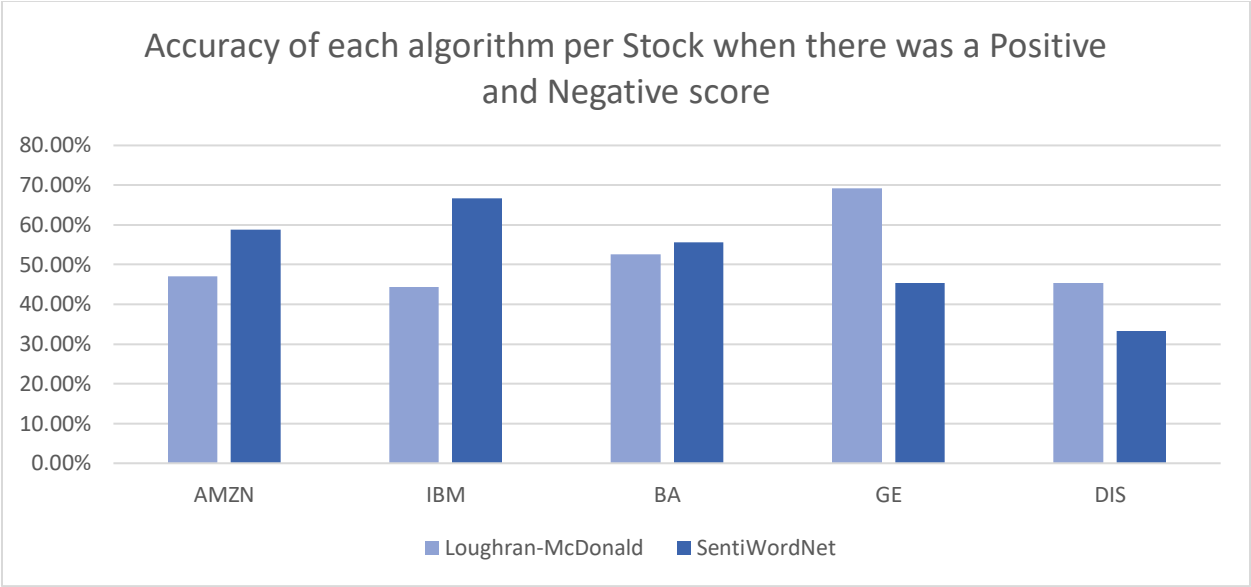




Mohammed Momin

negative_lm	positive_lm	negative_swn	positive_swn	Date	Price	lm_prediction	swn_prediction	actual
0	0	0	0	1998	31.568001	SELL	SELL	BUY
0	0	0	0	1999	32.122906	SELL	SELL	SELL
0	0	0	0	2000	30.704813	SELL	SELL	BUY
0	0	0	0	2001	31.074751	SELL	SELL	SELL
0	0	0	0	2002	21.574755	SELL	SELL	SELL
0	0	0	0	2003	17.904976	SELL	SELL	BUY
0	0	0	0	2004	24.534254	SELL	SELL	BUY
0.327076927	0.642830753	0.76631587	0.496621018	2005	26.803205	BUY	SELL	SELL
0.122771843	0.709651836	0	0	2006	24.40601	BUY	SELL	BUY
0.171162785	0.688964568	0	0	2007	33.728436	BUY	SELL	SELL
0.414197642	0.380386032	0.821833882	0.752997065	2008	30.32	SELL	SELL	SELL
0.556496613	0.508815624	1	1	2009	22.309999	SELL	SELL	BUY
0.845457133	0.937056649	0.987310998	0.784265261	2010	31.879999	BUY	SELL	BUY
0.932620885	0.925297923	0.966463843	0.989825597	2011	39.450001	SELL	BUY	BUY
0.978098355	0.981053879	0.981282491	0.969450728	2012	39.91	BUY	SELL	BUY
0	0	0	0	2013	50.580002	SELL	SELL	BUY
0	0	0	0	2014	75.389999	SELL	SELL	BUY
0	0	0	0	2015	94.25	SELL	SELL	BUY
0.475345008	0.44599284	1	1	2016	99.25	SELL	SELL	BUY
0.497881644	0.557955618	1	0	2017	108.980003	BUY	SELL	BUY
0.608973764	0.671374385	1	0	2018	111.620003	BUY	SELL	BUY





Conclusion

Both algorithms struggled to come up with positive or negative scores for some of the documents, but of the documents that received scores, the algorithms' accuracy ranged from 33% all the way up to 69%. There was no outright better algorithm. The Loughran-McDonald algorithm performed better for General Electric and Disney, but the SentiWordNet algorithm performed better for Amazon, IBM, and Boeing. While there were some promising results for specific stock-algo combinations, it might be mere coincidence. The accuracies ranging from 33-69% made me realized the accuracies are just on a distribution centered around 50%. Since there is already a 50-50 chance of a signal being right when random, each approach did not seem to effectively determine the outlook of a company.

There are a few ways to improve this project. First, would be to understand why some documents did not have positive and negative scores as that has decreased our sample size a

lot. I suspect it can be due to the api I used to retrieve the documents, or it could be due to the sentiments I left out of the algorithm. The second way to improve would be to include all the sentiments in the trading strategy instead of just positive and negative. Another way to expand the sample size would be to evaluate more stocks and more documents. For example, 10-Q documents, which are documents that would be filed like 10-K documents but every quarter instead of annually. Lastly, there can be different weights associated with various sections of the text depending on importance so the algorithm can determine which text to focus on when getting a sentiment score.

References

1. Baccianella, Stefano & Esuli, Andrea & Sebastiani, Fabrizio. (2010). "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.."
2. T. Loughran and B. McDonald, "Regulation and financial disclosure: The impact of plain English," J. Regul. Econ., vol. 45, no. 1, pp. 94–113, Nov. 2013.
3. T. Loughran and B. McDonald, "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks," Jan 2011.