Text Analysis & Modeling of Earnings Conference Call Transcripts

Group 7

Jian Tong Chua, Yunxuan Liao, Meilin Yuan, Jingchao Peng, Yanlin Zhang

Instructor: Dr. Patrick Houlihan

Course: QMSS-5067 Natural Language Processing

ABSTRACT

This paper explores the 2016-2020 earnings conference call transcripts from a few companies by applying a variety of text mining methods. Sentiment analysis using both traditional and domain-specific dictionaries shows an overall positive sentiment across years though it falls entering the coronavirus pandemic. Five different topic extraction models are used, where LDA and LSI perform relatively better in the overall coherence and partitioning transcripts into different clusters that each focus on part of the business value chain. We also try several supervised learning models to predict the percentage change of stock price based on the text corpus, while the best model returns a rather unsatisfactory predictive result that makes the model less useful.

1 INTRODUCTION

Financial market analysis always focuses on data from accounting, stock price, and other numerical data reported in P&L statements to extract signals. However, text-

based documents are also important signals for financial analysis. Recent advances in NLP methods for analyzing unstructured and text-based data at scale offer possibilities for understanding financial market behavior that could improve investments and market equity. Earning calls are one of those texts, which are periodic statements usually delivered by CEOs who attempt to influence investors' expectations of a company. Thus, in this project, we want to do some text analysis & modeling of earnings conference call transcripts.

We applied both supervised learning and unsupervised learning in our project to do sentiment analysis, topic analysis, and predictive analysis based on companies' earnings conference call transcripts.

2 LITERATURE REVIEW

There are lots of studies on text analysis and prediction of earnings conference call transcripts previously. ZhiqiangMa et al. (2020) used a deep learning framework, where an attention mechanism is applied to encode the text data into vectors for the discriminative network classifier to predict stock price movements. They found that the proposed model is superior to the traditional machine learning baselines and earnings call information can boost the stock price prediction performance. Sourav Medya et al. (2022) adopted pilot regression, GNN, and Doc2Vec methods to do sentiment analysis and stock price movement prediction. They find that the semantic characteristics of transcripts are relatively more accurate to predict stock price movements, and also

establish that the transcripts have more predictive power than traditional hard data such as actual and estimated values of sales and earnings per share.

Past studies also indicated that topic modeling and sentiment analysis were also helpful in finding new insights from earrings conference call transcripts. Huang et al. (2015) used Latent Dirichlet Allocation (LDA) to go beyond traditional analysis of “how tests are being said” to “what is being said” in text analysis. This study found that “the proportion of new information in analyst reports relative to that in conference calls increases when firms face a higher level of competition, have a greater litigation risk, or operate in a more volatile information environment” (Huang et al., 2015), which is a very useful piece of information when trying to interpret a firm's situation through conference calls. As for sentiment analysis, Noor Malik (2020) conducted a study on using different neural network models to predict sentiments on future outlooks through conference calls. Through this research, Noor Malik was able to quickly identify and predict sentiments that provide insights into a firm's future performance, and the research also suggested that Nearest Centroid, though being the simplest model, had the best accuracy score of 82%, which could be used as a reference for our study.

3 DATA

The data we used is called “Stock Values and Earnings Call Transcripts: a Sentiment Analysis Dataset”. The authors of this dataset are Dexter Roozen and Francesco Lelli from Tilburg University. This dataset is a collection of 188 earnings call transcripts, 11970 related stock prices, and 1196 sector index values in the period 2016-

2020 related to the NASDAQ stock market. The data collection was done by Yahoo Finance (enabled the search for stock values) and Thomson Reuters Eikon (provided the earnings call transcripts). In the downloaded dataset folder, there are ten Microsoft Excel Comma Separated Values Files from NASDAQ companies, one Microsoft Excel Comma Separated Values File from the NASDAQ sector index, and 188 Text Documents from NASDAQ companies. (Citation & Publication: Roozen, D.; Lelli, F. Stock Values and Earnings Call Transcripts: a Dataset Suitable for Sentiment Analysis. Preprints 2021, 2021020424 (DOI: 10.20944/preprints202102.0424.v1))

4 METHODOLOGY

For the general data preprocessing, we did the following:

1) Removed the punctuations, digits, and special characters

2) Trimmed the white spaces and converted all the remaining words into their lowercase forms

3) Removed stopwords using the NLTK API and added some additional earnings call-specific stopwords including presentation, operator, year, quarter, copyright, thanks, etc;

4) Lemmatization using Wordnet Lemmatizer.

4.1 EXPLORATORY TEXT ANALYSIS

19 preprocessed earning call documents from Jan. 2016 to July 2020 for Apple, Google, and Amazon were used for exploratory text analysis. Five word clouds were generated corresponding to the combined corpus from each year and four frequent terms

were selected based on the word clouds, then their term frequencies per document were plotted against time to visualize the trend across quarters and years.

4.2 SENTIMENT ANALYSIS

We computed sentiment polarity scores with regard to each document in the dataset by three different dictionaries and two sentiment APIs.

4.2.1 SENTIMENT DICTIONARIES

Bing sentiment lexicon is a general-purpose English sentiment lexicon that categorizes words in a binary fashion, either positive or negative.

The Loughran-McDonald Master Dictionary was initially developed in conjunction with our paper published in the Journal of Finance (When is a Liability not a Liability?, 2011). The dictionary provides a means of determining which tokens (collections of characters) are actual words, which is important for consistency in word counts.

Elaine Henry Dictionary is a method from the article “Are Investors Influenced By How Earnings Press Releases Are Written” which was published in The Journal of Business.

4.2.2 SENTIMENT APIS

TextBlob is a library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

VADER is a less resource-consuming sentiment analysis model that uses a set of rules to specify a mathematical model without explicitly coding it. VADER consumes fewer

resources as compared to Machine Learning models as there is no need for vast amounts of training data.

4.3 TOPIC MODELLING

A topic model is to extract the abstract “topics” that occur in a collection of documents. It’s a popular text-mining tool for discovering hidden semantic structures. For our purposes, we used a few different topic models to apply to the whole corpus (n = 188 earnings call transcript documents) in order to explore how top tech companies’ earning call documents could be partitioned into different clusters, as well as the focus of each cluster. Some of them might be more about stressing market expansion or launching new products, while others might focus on core technologies, merging & acquisition, fiscal trends, and so on.

We deployed five different methods/models for extracting topics from the corpus and assigning them to our original transcript documents. Two of them (manual method and ordinary LDA) were applied/tuned in detail, while the rest of the three (eLDA, LSI, and HDP) were more of an experimentation purpose.

4.3.1 MANUAL METHOD

The idea of this manual method is to find some numeric representation of both topics and documents, and then see which topic fits each of our documents the best. Both parts are highly subjective and there is lots of room for discussion though. Our method here was to

1. Compute the wording embedding matrix using the **Word2Vec** model

Word2Vec is an NLP technique that uses neural networks to learn word associations from a large corpus of text. Each word in the corpus could be represented by an array of numbers, and word vectors collectively form a vector space called wording embeddings where words that share similar meanings in the context are numerically closer to each other, and vice versa.

We specified 500 word vectors in our word embeddings as the model input and computed the mean of the word vectors in each document to return a numeric representation of the document.

2. Specify eight topics by querying eight different words in the Word2Vec model

One of the Word2Vec model methods is to query a word to return a vector corresponding to that word's numeric representation, which we used here as a way to specify topics. The choice of words is completely subjective depending on what range of words we consider are good summaries of the earning call documents. We chose "youtube", "fiscal", "use", "revenue", "develop", "expand", "tech", and "product" to query and fetch their vectors.

3. Compute **cosine similarities** between documents and topics

This step is rather straightforward: to compute the cosine similarities between all 188 documents and eight topics and see which topic has the highest cosine similarity in each document so that we are able to assign that topic to the document.

4.3.2 LATENT DIRICHLET ALLOCATION

Latent Dirichlet Allocation (LDA), one of the most frequently used algorithms in topics modeling, is a generative probabilistic model for collections of discrete data. LDA

is a three-level hierarchical Bayesian model, where each item of a collection is modeled as a finite mixture attributable to one of the underlying sets of latent topics. After pre-specifying the number of topics before deploying it, LDA returns the sorted words in each topic with respect to their probability score.

There are three hyperparameters to tune in the ordinary LDA model: **k**, **alpha**, and **beta**. K is the number of topics required to be specified before model deployment, and we usually find the optimal k by enumerating multiple LDAs and seeing which one returns the highest coherence score. Topic coherence is defined as the measure of scoring a single topic by measuring the degree of semantic similarity between high-scoring words in the topic. Here the "c_v" coherence measure is used, which, by definition, is based on a sliding window, one-set segmentation of the top words, and an indirect confirmation measure that uses normalized pointwise mutual information and the cosine similarity. The higher the coherence score an LDA model returns, the more coherent, and usually the better the model is in clustering groups of topics.

Similar to the elbow curve of choosing the optimal k in the context of k-means clustering, the elbow curve of LDA visualizes the trend of changing coherence score when the number of topics increases. It makes sense to pick the k with a relatively higher coherence score (not necessarily the highest) but also marks an "elbow" turning point where the topic coherence curve starts to flatten out. If some keywords appear repetitively in multiple topics, then the LDA might allocate too many topics even if the coherence score is rather high. An optimal LDA model ideally should be both coherent and yet have distinct words in each topic it partitions.

Besides, if an LDA has symmetric distribution, alpha represents the document-topic density, and beta represents the topic-word density. A higher alpha means each document consists of more topics and vice versa. A higher beta means our topics are made of most words in the corpus and vice versa. If an LDA has asymmetric distribution, then higher alpha results in a more specific topic distribution per document, while higher beta leads to a more specific word distribution per topic.

There is some very general rule of thumb of whether alpha and beta should be higher or lower, but it's really context-based and determined empirically. Our simple method to tune hyperparameters here is to use the k we determined based on the elbow curve and grid search a combination of different alpha and beta in the range of (0.01, 1, 0.3) and see which tuple generates the highest coherence score.

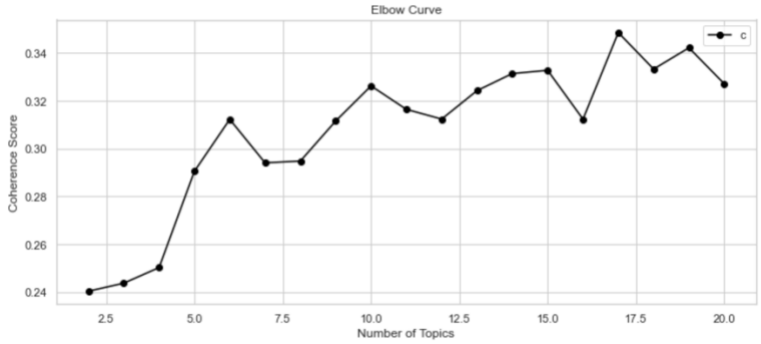

Figure 1. LDA Elbow Curve

	Validation_Set	Topics	Alpha	Beta	Coherence
59	100% Corpus	10	asymmetric	symmetric	0.326183
54	100% Corpus	10	symmetric	symmetric	0.326183
4	75% Corpus	10	0.01	symmetric	0.326183
39	100% Corpus	10	0.31	symmetric	0.326183
34	100% Corpus	10	0.01	symmetric	0.326183
9	75% Corpus	10	0.31	symmetric	0.326183
29	75% Corpus	10	asymmetric	symmetric	0.326183
24	75% Corpus	10	symmetric	symmetric	0.326183
20	75% Corpus	10	symmetric	0.01	0.321886
55	100% Corpus	10	asymmetric	0.01	0.321886
50	100% Corpus	10	symmetric	0.01	0.321886
35	100% Corpus	10	0.31	0.01	0.321886
25	75% Corpus	10	asymmetric	0.01	0.321886
0	75% Corpus	10	0.01	0.01	0.321886
30	100% Corpus	10	0.01	0.01	0.321886
5	75% Corpus	10	0.31	0.01	0.321886
44	100% Corpus	10	0.61	symmetric	0.320453
14	75% Corpus	10	0.61	symmetric	0.320453
15	75% Corpus	10	0.91	0.01	0.319993
45	100% Corpus	10	0.91	0.01	0.319993

Table 1. LDA Tuning Result

Based on the tuning results above, optimal k = 10, optimal alpha = 0.01, and optimal beta = ‘symmetric’.

4.3.3 ENSEMBLE LDA

Ensemble LDA (eLDA) is constructed on top of the idea of LDA but works somewhat differently. It generates more stable topics from the results of multiple LDAs and will remove topics from the result that are noise and not reproducible. The idea of trying eLDA is based on the possibility that different LDAs are trained on the same corpus with random seeds and a different number of topics, and the need for more reliable topics for classifying new documents or for other supervised learning purposes.

Ensemble LDA works by

1) Train an ensemble for multiple LDAs
2) Group topic clusters together via a variation of the DBSCAN algorithm
3) If the topic cluster is large enough, keep the core of clusters as reliable topics and discard the rest

4.3.4 LATENT SEMANTIC INDEXING

Latent Semantic Indexing (LSI), yet another popular topic model, is an indexing and retrieval method that uses dimensionality reduction technique to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. We tried LSI because the traditional document-term matrics as a topic model input are sparse and noisy, and this kind of vector space representation cannot handle the problems of synonymy (different words having the same meaning) and polysemy (a word having multiple meanings).

LSI works by

1) Convert the text corpus to a document-term matrix
2) Apply truncated Singular Vector Decomposition (SVD) to reduce the dimensions, noise, and cardinality of the matrix
3) Encode words/documents with extracted topics

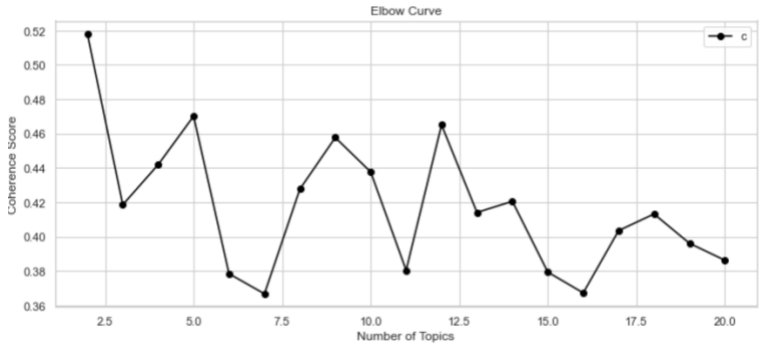

Figure 2. LSI Elbow Curve

Based on the elbow curve above, we choose the optimal number of topics = 5.

4.3.5 HIERARCHICAL DIRICHLET PROCESS

Hierarchical Dirichlet Process (HDP) is a non-parametric Bayesian approach to clustering grouped data. One advantage of HDP is that it uses Bayesian update to infer the number of topics so that we don't have to specify it as the model input beforehand. The gensim HDP API provides the speed of online variational Bayes with modeling flexibility. The idea behind Online Variational Bayes in general is to optimize the variational objective function with stochastic optimization.

4.3.6 TOPIC ASSIGNMENT

Aside from the manual method that assigns the topic based on the highest cosine similarity score with each document, the rest of the four algorithms all use some other way:

1) Look at the coherence loading scores for the best words based on the printed topics

2) Choose a topic with the highest score

Note that the raw output of all printed topics where each word has its loading scores can be found in the Appendix.

4.4 SUPERVISED LEARNING

In addition to topic modeling, we also perform an attempt to predict the 1-period stock return post earning calls using several commonly used supervised learning approaches. We approached this task of predicting the 1-period stock return as a regression problem, with the target variable being the 1-day stock return in percentage terms as follows:

$$return_t = \frac{Price_t}{Price_{t-1}} - 1$$

Since these earning calls could be either before or aftermarket trading hours, we also had to extract the date and time the earning call occurred which was conveniently included in the call transcripts. Using the time of the calls, we could then identify the previous period price as the last daily closing price prior to the call and the next period price as the daily closing price of the stock immediately after the call.

The only predictors we used are the features extracted from the text body, on which we first did several preprocessing steps. Firstly, we discarded the irrelevant parts of the transcript like the legal definitions and disclaimers, and the names of the participants as these pieces of text were standardized across all earning calls and did not provide much information for modeling and prediction purposes. Next, we cleaned up the remaining text, including removing stopwords, numbers, and symbols, and applied stemming. Vectorize

via Term Frequency - Inverse Document Frequency (TF-IDF) using Scikit Learn's TfIdfVectorizer function. For each document, we used the frequencies of the top 1000 unigrams, bigrams, and trigrams as inputs to our model.

We experimented with 5 different machine learning models, over three different preprocessing strategies (unigrams, bigrams, trigrams) for a total of 15 different models. The models were trained on 150 of the 188 transcripts in our data set with the remaining 38 set aside as a test set. For each model, hyperparameters were tuned using a grid search approach. This consisted of identifying a set of different possible combinations of hyperparameters for each model and choosing the best performing set on basis of the cross-validated mean absolute error. The below table shows the 5 different models used and their hyperparameters that we tuned over.

Hyperparameters Tuned				
Kneighbors Regressor	RandomForest Regressor	SupportVector Regressor	Huber Regressor	GradientBoosting Regressor
n neighbors	n estimators	kernel C	epsilon Alpha	n estimators learning rate max depth

Table 2. Predictive Model Hyperparameters

5 RESULTS & DISCUSSION

5.1 EXPLORATORY TEXT ANALYSIS

Figure 3. Apple Word Clouds over time

For Apple, the most common term through time is “iPhone”, which suggests that it has always been the most important product the company has been focusing on. There were some minor changes in term appearance that shows how the company has shifted its focus in different years, such as there is “store” in 2017 which indicated that Apple was focusing on building off-line stores, and “china” in 2018 when “Apple” was expanding its market to China.

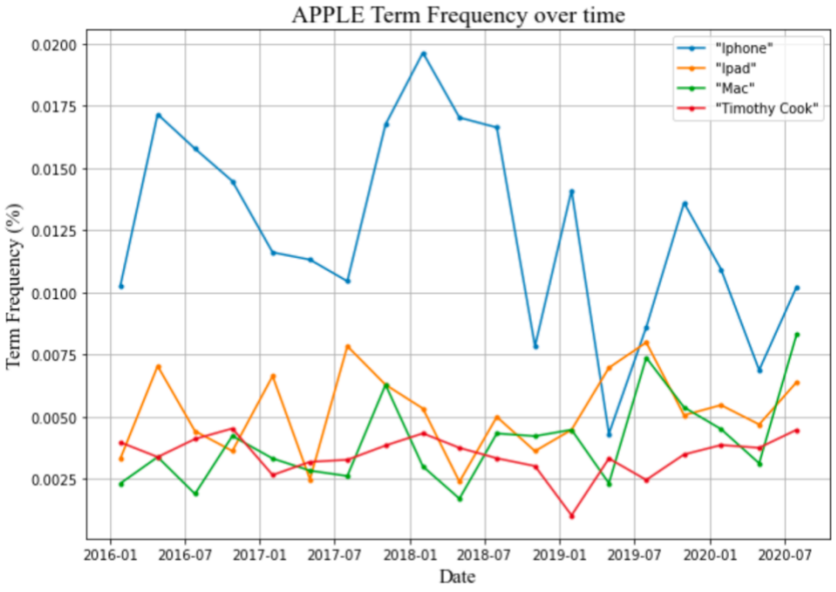

Figure 4. Apple Term Frequency over time

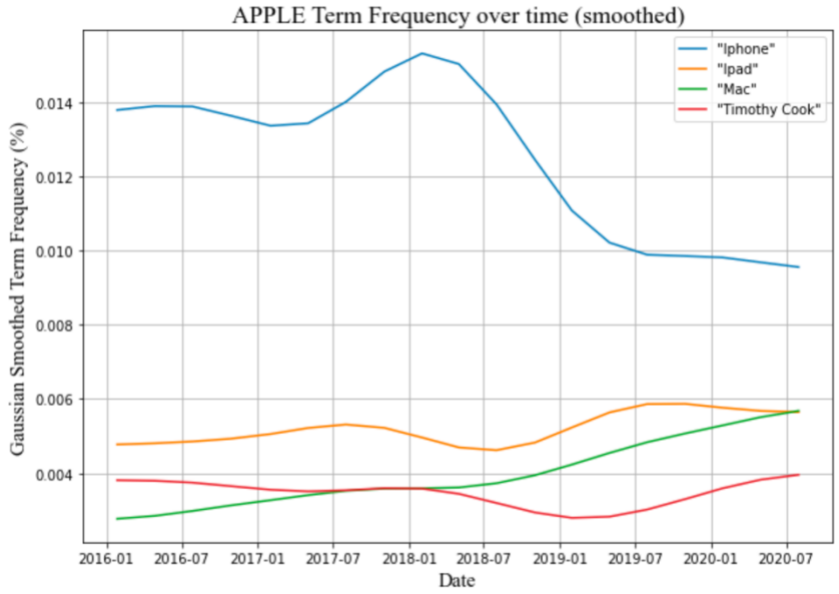

Figure 5. Gaussian-smoothed Apple Term Frequency over time

When looking at the four most frequent terms' term frequency per document, it was identified that "iPhone" appeared way more than the other 3 terms in 2016, but there is a

trend that shows it started to decrease in 2018, which was also the time that iPhone reached its peak in the market. In recent years, “mac” is starting to catch up with “iPhone”, suggesting that Apple might be shifting focus.

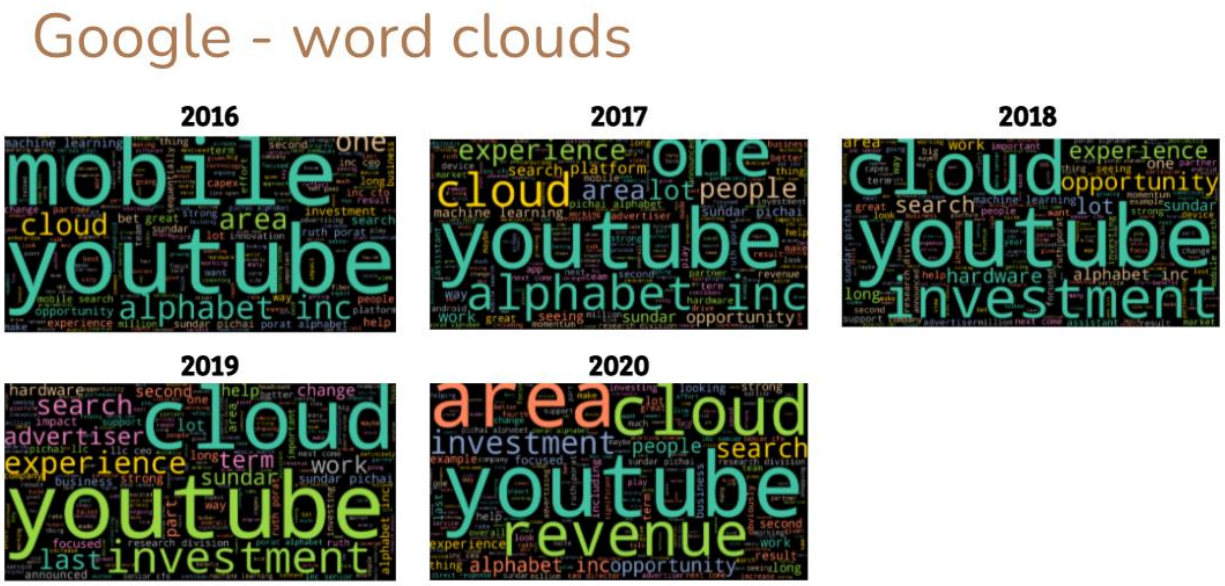

Figure 6. Google Word Clouds over time

For Google, “youtube” has always been the most popular term. It can also be seen that “cloud” was not a big term in 2016 but it slowly increased its importance during conference calls and it is now one of the big terms for Google. Overall, Google has also been shifting from focusing on “investment” in 2018 and 2019 to focusing on “revenue” in 2020.

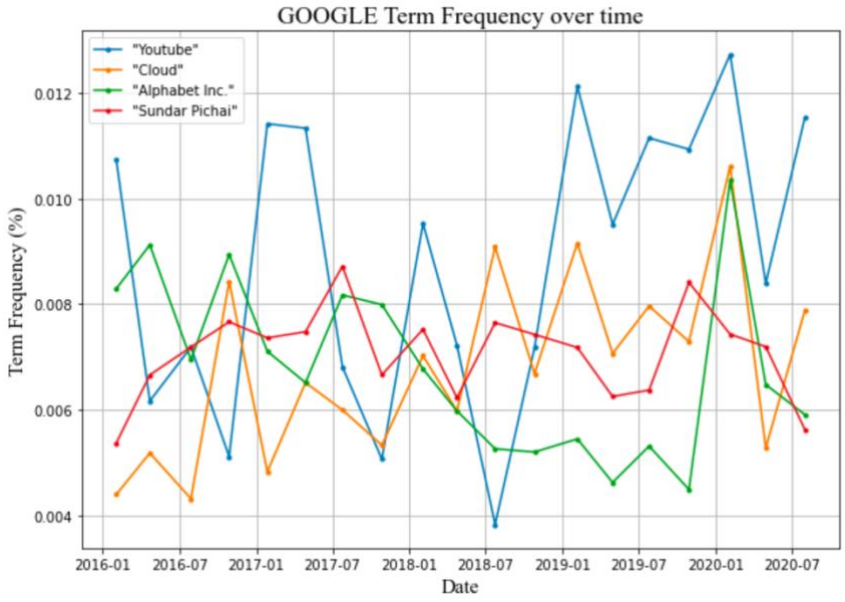

Figure 7. Google Term Frequency over time

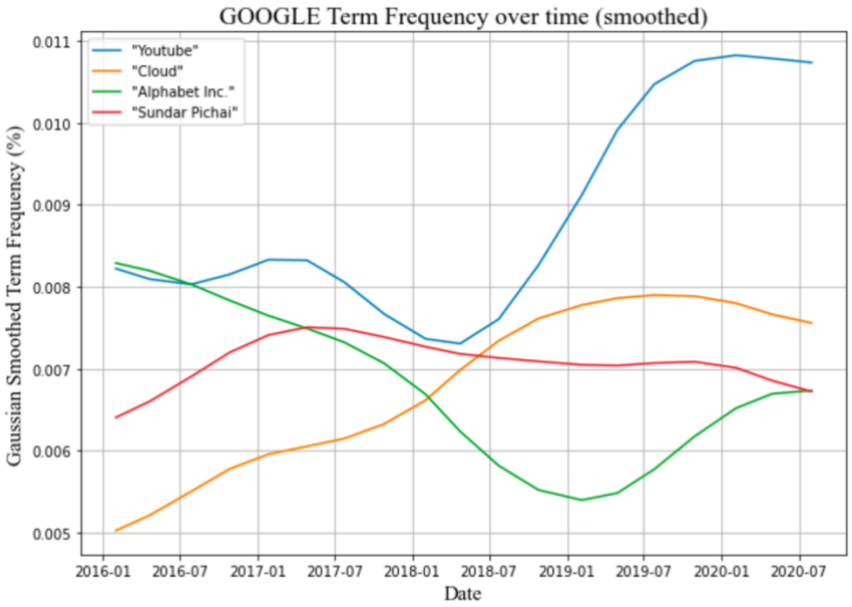

Figure 8. Gaussian-smoothed Google Term Frequency over time

Compared to Apple, Google's most frequent terms did not start off with huge differences in frequencies back in 2016. In fact, all the terms were about the same in 2018,

it is only in recent years that “youtube” is gaining more frequency. This suggests that Google has a more diverse portfolio of its most important businesses.

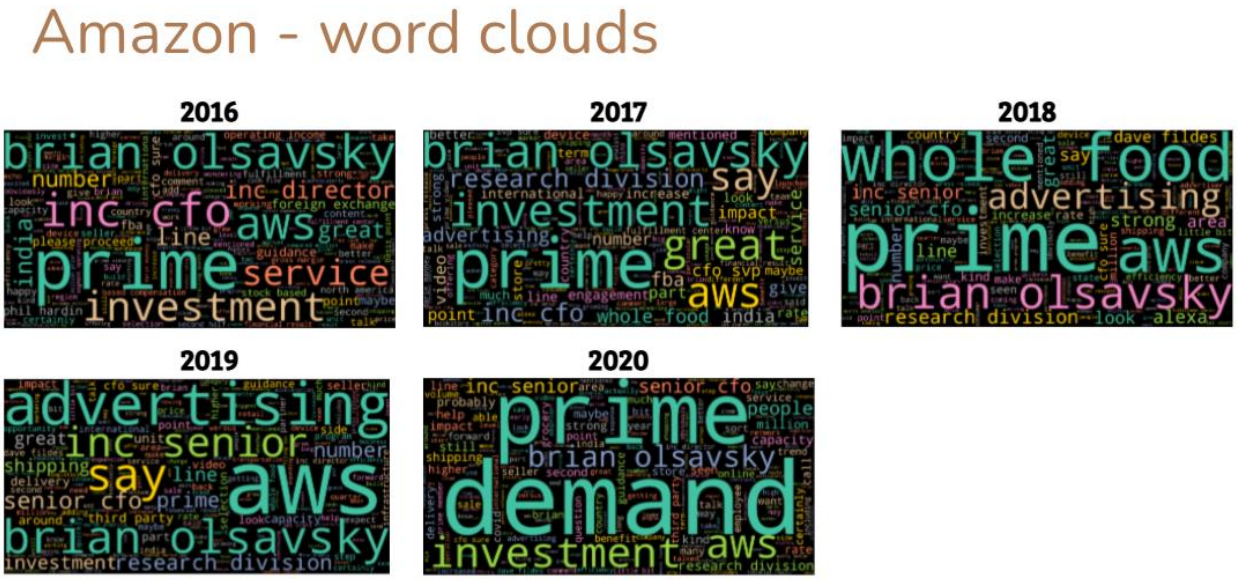

Figure 9. Amazon Word Clouds over time

As Apple and Google’s most frequent terms were mostly product-oriented, Amazon is relatively more person-oriented. The name “Brian Olsavsky” appeared multiple times as one of the important terms of the year. This insight shows that as the CFO of amazon, “Brian Olsavsky” is really important to the company.

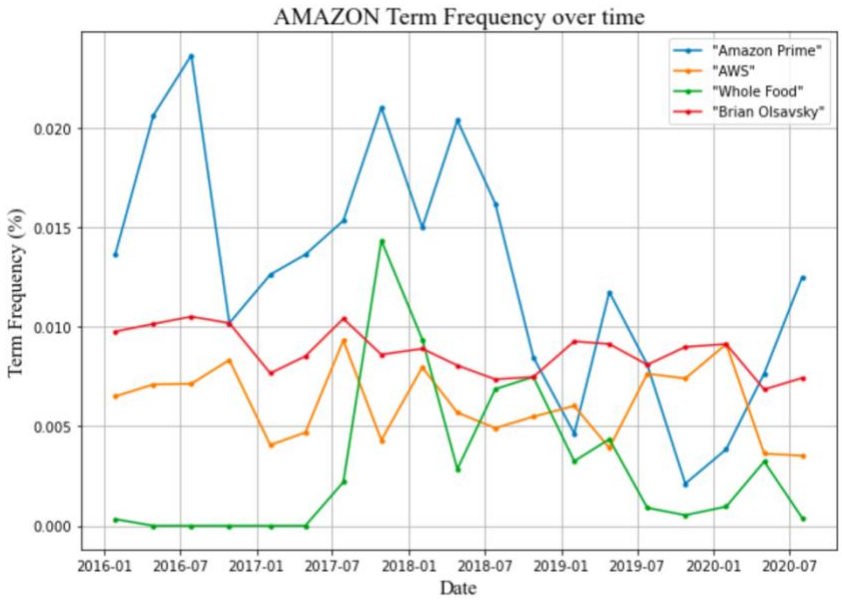

Figure 10. Amazon Term Frequency over time

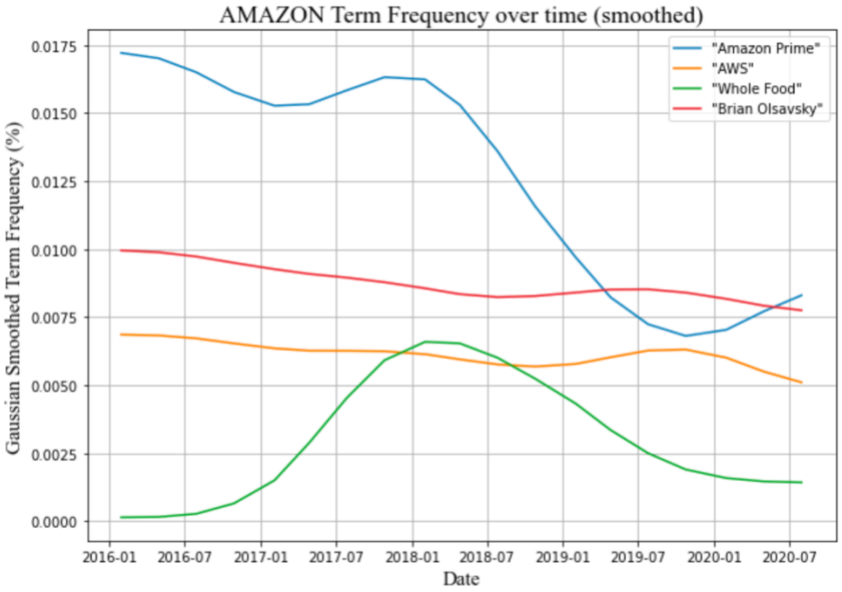

Figure 11. Gaussian-smoothed Term Frequency over time

There is a huge gap between the most frequent and least frequent terms in 2016, but the gap is slowly decreasing until 2020. This phenomenon is similar to that of Apple,

as “Amazon Prime " was also one of the most popular products, but it is starting to be overweighed by emerging businesses, especially during the years from 2019 to 2020. Interestingly, “Brian Olsavsky” has always been in a stable trend, with no big fluctuations.

5.2 SENTIMENT ANALYSIS

We applied five methods for computing sentiment polarity scores on earnings call documents from Apple, Google, and Amazon.

The table shows the descriptive stats of all five sentiment dictionaries. Textblob and VADER scores are just generated via their APIs, and the rest of the three were done manually using the function we did in HW3.

	bing_senti	vader_senti	tb_senti	lm_senti	henry_senti
count	19.000000	19.000000	19.000000	19.000000	19.000000
mean	0.365038	0.999926	0.152786	0.497747	0.712092
std	0.045521	0.000045	0.013722	0.121617	0.098764
min	0.301961	0.999900	0.131759	0.169492	0.506494
25%	0.324141	0.999900	0.141236	0.437218	0.657387
50%	0.354839	0.999900	0.150431	0.515625	0.723077
75%	0.401937	0.999950	0.164440	0.557778	0.766434
max	0.454545	1.000000	0.178616	0.729730	0.921053

Table 3. Descriptive Statistics of Sentiment Polarity Scores

5.2.1 Sentiment Polarity Over Time

Here are three graphs comparing how 4 different sentiment scores change over time for Apple, Google, and Amazon respectively.

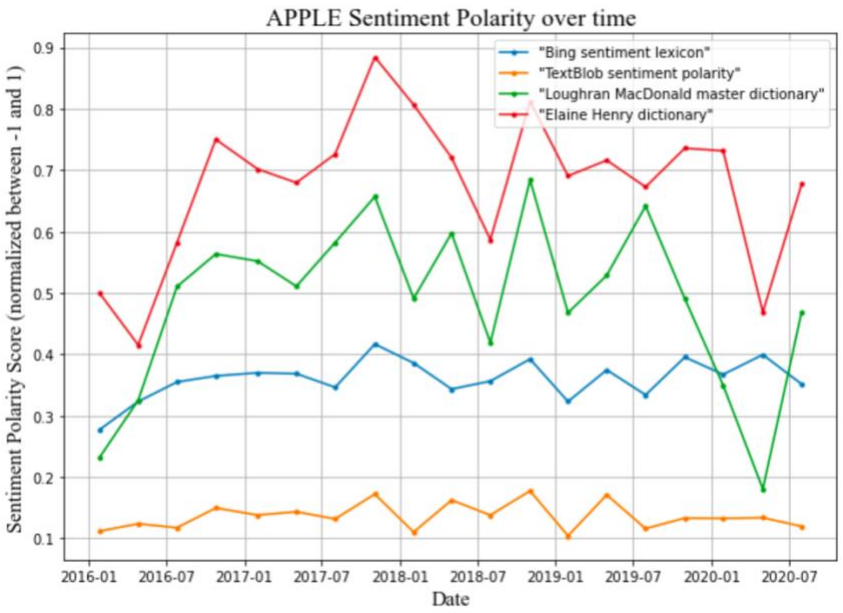

Figure 12. Apple Sentiment Polarity over time

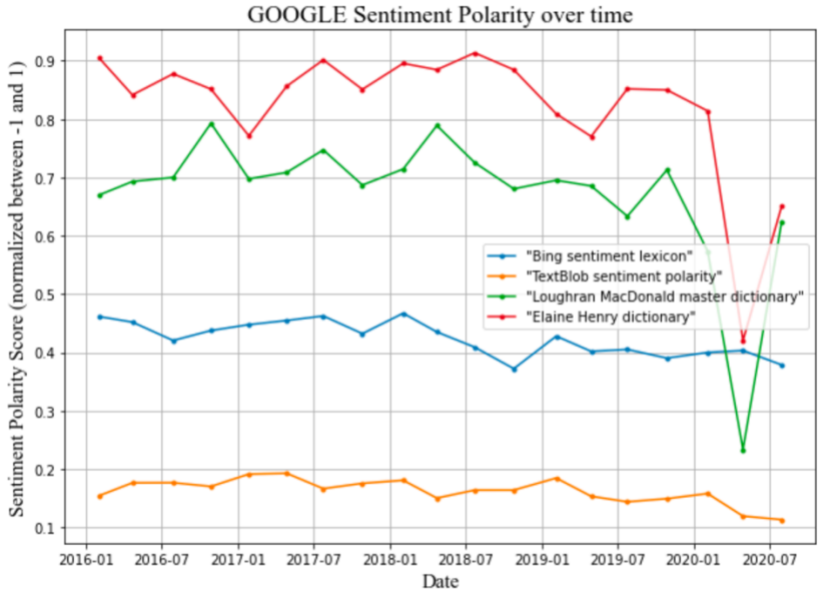

Figure 13. Google Sentiment Polarity over time

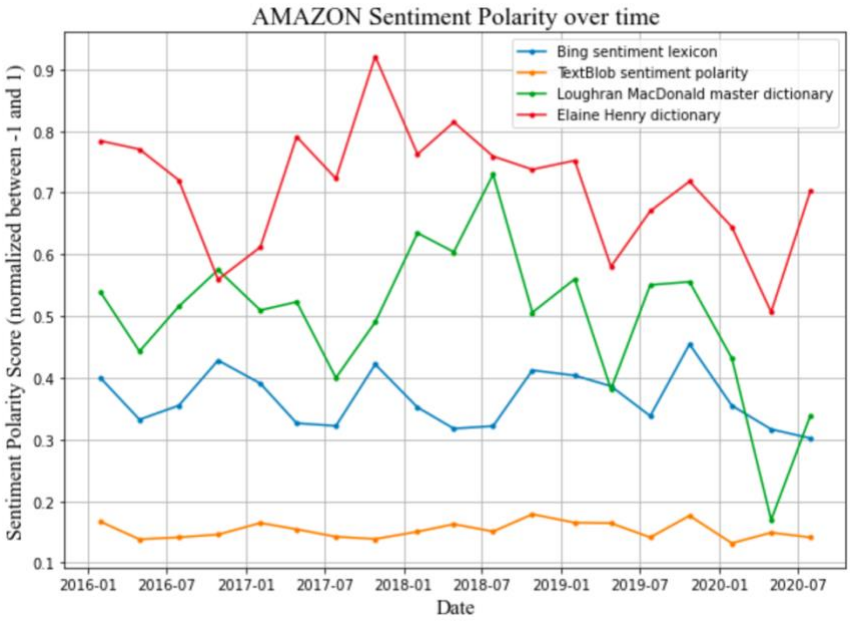

Figure 14. Amazon Sentiment Polarity over time

We can see that the McDonald and Elaine Henry sentiments share similar sentiment trends in that the ups and downs are significant. While the results of the other two methods are relatively stable here. VADER was discarded since it doesn't make sense in our earning call corpus.

5.2.2 Loughran McDonald Sentiment Comparison

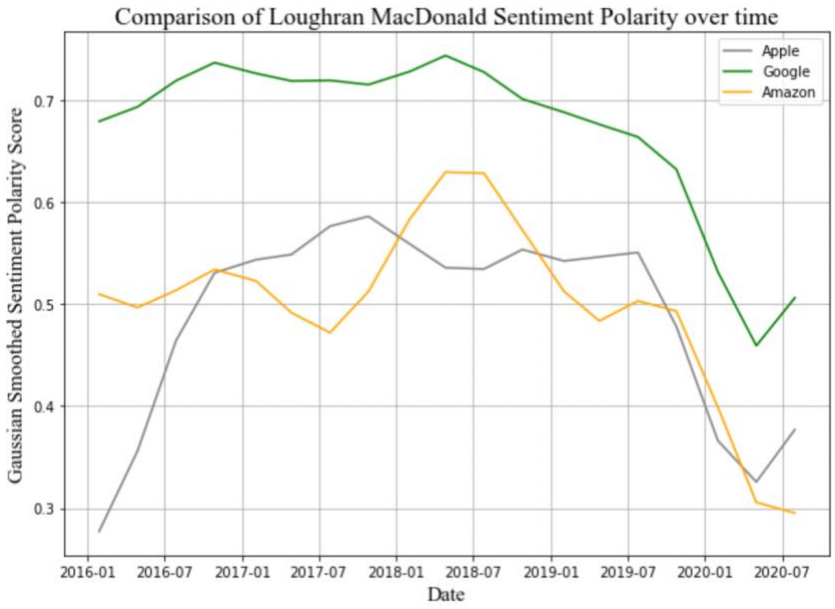

Figure 15. Comparison of Loughran-McDonald Sentiment Polarity

Here, we compared three companies' MacDonald sentiment scores over time. The curves are all gaussian-smoothed. Overall, during the past several years, Google has had the highest sentiment polarity, while Apple and Amazon are intertwined. It seems that these three companies had a lower sentiment, though still fairly positive, when Covid-19 came.

5.3 TOPIC MODELING

5.3.1 MANUAL METHOD

	Topic	Documents
0	Topic 1: Youtube	[2, 7, 9, 11, 14, 15]
1	Topic 2: Fiscal	[0, 6, 8, 19, 20, 21, 22, 24, 25, 29, 33, 38, 39, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 74, 75, 76, 77, 114, 115, 116, 117, 118, 119, 122, 123, 124, 125, 126, 127, 128, 129, 131, 132, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168]
2	Topic 4: Revenue	[1, 5, 13, 17, 23, 26, 27, 28, 30, 31, 32, 34, 35, 36, 78, 120, 121, 133, 134, 135, 136, 137, 138, 139, 140, 141, 143, 147, 150]
3	Topic 5: Development	[169, 170, 174]
4	Topic 6: Expansion	[142, 144, 145, 146, 148, 149, 151, 172]
5	Topic 7: Technology	[3, 4, 10, 12, 16, 18, 37, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 73, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 130, 171, 173, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187]

Table 4. Manual Method Topic Assignments

The table above shows the final topic assignment with regard to all documents in the corpus using the manual Word2Vec-cosine-similarity method. Two out of eight topics, “customer” and “product”, are not assigned to any documents, perhaps because those two words are too frequent in the corpus to stand out as unique topics. Numerically speaking, the average similarities between those two words and 188 documents are among the lowest. The distribution of topic assignments is not rather uniform. Those labeled with “Fiscal” and “technology” collectively make up most documents from our corpus, whereas “Youtube”, “development”, and “expansion” are each allocated to less than 10 documents.

5.3.2 LATENT DIRICHLET ALLOCATION

Before jumping to the topic assignment result, we used an API called pyLDAviz that visualizes the LDA result on a PC1-PC2 2-dimensional space, where the size of the bubbles (topics) corresponds to the size of the corpus, and the positions of the bubbles show how relevant or irrelevant different topics are to each other. On the right-hand side, the top-10 most salient words are listed. If no bubble is selected, then the blue bar

represents the overall term frequency. If a bubble is selected, then the red bar gives the estimated number of times a given word was generated by a given topic. I took two screenshots of the pyLDAviz output, each matches topics 2 and 3 respectively.

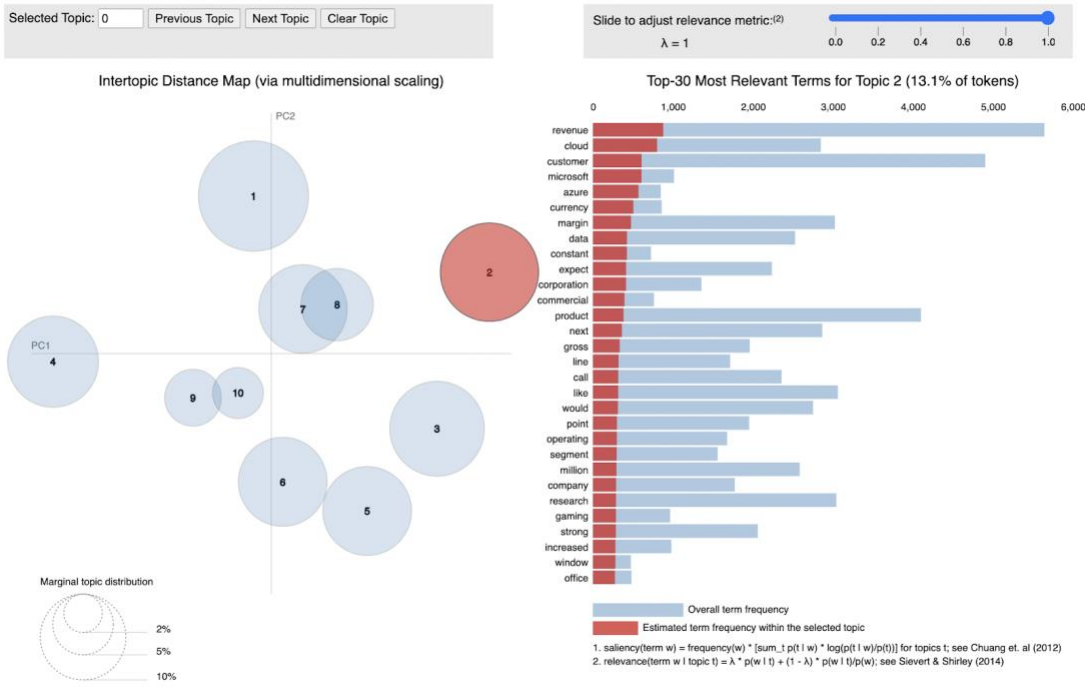

Figure 16. pyLDAviz output of topic 2

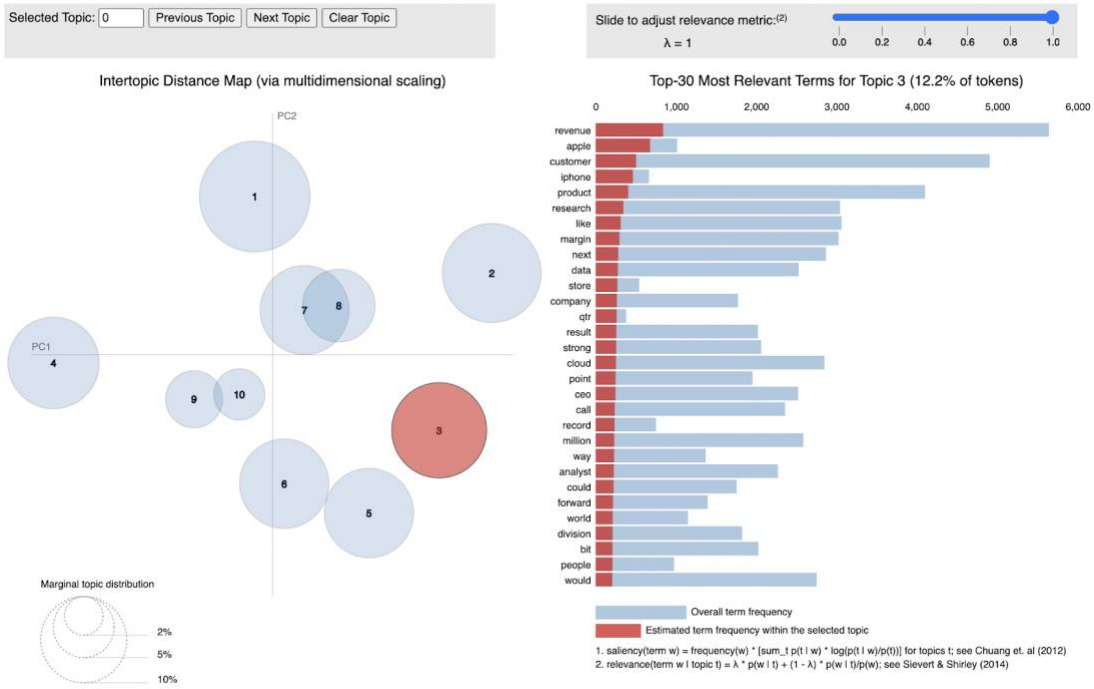

Figure 17. pyLDAviz output of topic 3

Topic 1: Microsoft product
Topic 2: Microsoft cloud computing
Topic 3: Apple product & tech
Topic 4: ASML EUV (more revenue-oriented)
Topic 5: Amazon product & tech
Topic 6: Cisco software and products
Topic 7: Tech center (cisco)
Topic 8: Cloud, tech, software
Topic 9: ASML EUV (more customer-oriented)
Topic 10: ASML EUV (more research-oriented)

topic	Documents
Microsoft product & tech	[9, 139, 145, 147]
Microsoft cloud computing	[14, 25, 133, 134, 135, 136, 137, 138, 140, 141, 142, 143, 144, 146, 148, 149, 150, 151, 171, 180]
Apple product & tech	[6, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 44, 57, 58, 59, 61, 63, 67, 68, 70, 71, 72, 74, 108, 115, 116, 117, 118, 119, 120, 121, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 165, 166, 167, 170, 172, 173, 176, 178, 179]
ASML EUV (more revenue-oriented)	[13, 42, 43, 45, 47, 48, 49, 51, 60, 62, 64, 65, 66, 69, 73, 75, 79, 81, 83, 88, 103, 105, 106, 174]
Amazon product & tech	[1, 12, 15, 41, 50, 52, 53, 55, 76, 78, 80, 82, 84, 85, 86, 87, 89, 90, 92, 95, 97, 98, 99, 100, 101, 102, 104, 109, 110, 112, 114, 122, 123, 125, 169, 182]
Cisco software and products	[3, 8, 10, 11, 17, 46, 77, 91, 93, 94, 96, 107, 113, 177, 185, 186]
Tech center (cisco)	[0, 2, 4, 7, 18, 54, 111, 124, 126, 127, 128]
Cloud, tech, software	[5, 56, 175, 187]
ASML EUV (more customer-oriented)	[16, 129, 130, 131, 132, 164, 168, 181]
ASML EUV (more research-oriented)	[183, 184]

Table 5. LDA Topic Assignments

We summarized the topics based on the raw printed topics. We also assigned all ten topics to our documents which is shown in the table on the right. There are more topics clustering around the theme of Apple, ASML, and Amazon, and their technologies and products accordingly. Note that not all topics are allocated with respect to each

company in a clear-cut manner. There are multiple topics that are loosely relevant to ASML but stress different parts of the businesses/value chains.

5.3.3 ENSEMBLE LDA

topic	Documents
customer, product, revenue, research, etc	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187]

Table 6. Ensemble LDA Topic Assignments

The eLDA only returns one topic, which means only one stable topic is a good representation of tech companies' earning call text. Therefore, eLDA is not particularly informative since it does not partition our documents at all.

5.3.4 LATENT SEMANTIC INDEXING

Topic 1: product and R&D

Topic 2: ASML EUV + Microsoft Azure (more tech-oriented)

Topic 3: Micron EUV Microsoft (more product-oriented)

Topic 4: Micron Nvidia (more fiscal-oriented)

Topic 5: Microsoft Micron Nvidia commercials

topic	Documents
ASML EUV + Microsoft Azure (more tech-oriented)	[5, 7, 8, 9, 10, 11, 12, 16, 18, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 119, 120, 121, 122, 123, 127, 133, 139, 141, 144, 146, 147, 153, 155, 156, 158, 159, 161, 162, 167, 168]
Micron EUV Microsoft (more product-oriented)	[0, 1, 2, 3, 4, 6, 13, 14, 15, 17, 35, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 134, 135, 136, 137, 138, 140, 142, 143, 145, 148, 149, 150, 151]
Micron Nvidia (more fiscal-oriented)	[169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187]
Microsoft Micron Nvidia commercials	[19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 118, 124, 125, 126, 128, 129, 130, 131, 132, 152, 154, 157, 160, 163, 164, 165, 166]

Table 7. LSI Topic Assignments

Only four out of five topics generated by LSI were assigned. The topic discarded might have a too broad focus as it talks about products and R&D in general. The rest of the topics I summarized are loosely divided into ASML & Microsoft technologies, Microsoft & Micron products, Micron & Nvidia fiscal trends, and Microsoft & Micron & Nvidia commercials. The assignment is quite even and roughly focuses on different parts of the value chains.

5.3.5 HIERARCHICAL DIRICHLET PROCESS

Topic 1: Microsoft product & data
Topic 2: Apple product R&D
Topic 3: Micron & Cisco fiscal
Topic 4: Amazon product
Topic 5: Micron & Cisco product
Topic 6: Cloud computing & research platform
Topic 7: Micron tech NAND
Topic 8: Microsoft executive (more customer-oriented)
Topic 9: Apple & ASML product
Topic 10: Microsoft executive (more research-oriented)
Topic 11: Cisco & Microsoft tech
Topic 12: ASML executive
Topic 13: AMD & Microsoft
Topic 14: Apple product
Topic 15: ASML tech
Topic 16: ASML tech & executive
Topic 17: ASML & Nvidia
Topic 18: Amazon product & executive
Topic 19: Amazon executive
Topic 20: Apple marketing

topic	Documents
Microsoft product & data	[1, 4, 6, 7, 8, 10, 11, 12, 13, 14, 17, 29, 38, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 59, 63, 65, 68, 69, 70, 76, 77, 78, 80, 81, 83, 89, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 115, 125, 128, 129, 130, 131, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 157, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 184, 185, 186]
Apple product R&D	[0, 2, 3, 5, 9, 15, 16, 18, 22, 25, 26, 30, 35, 37, 39, 40, 58, 61, 62, 64, 67, 71, 72, 75, 79, 82, 84, 85, 86, 88, 91, 92, 93, 94, 114, 116, 122, 127, 155, 156, 158, 183, 187]
Micron & Cisco fiscal	[19, 20, 21, 24, 27, 31, 34, 66, 73, 74, 87, 90, 118, 119, 120, 121, 123, 124, 126, 132]
Amazon product	[23, 28, 33, 36, 60, 117]
Micron & Cisco product	[32]

Table 8. HDP Topic Assignments

HDP notably generated 20 topics in total, but only five of them were assigned to our transcripts. A majority of them are assigned to the topic of “Microsoft product & data”, while the rest of them are distributed to “Apple products + R&D”, “Micron & Cisco fiscal trends”, “Amazon products”, and “Micron & Cisco products”. The allocation is fairly skewed, and the topics assigned don’t differ from each other rather clearly.

5.4 SUPERVISED LEARNING MODEL PREDICTIONS

In total, we experimented with 15 different models (5 algorithms by 3 different preprocessing approaches). Hyperparameter for each model was tuned over a grid of possible values and the best performing model was chosen. The cross-validated mean absolute errors are tabulated below.

Model	Mean Absolute Error		
	Unigram	Bigram	Trigram
KNN	0.05115	0.05258	0.05295
Random Forest	0.05403	**0.05049**	0.05326
Support Vector Machine	0.05429	0.05432	0.05534
Huber Regressor	0.05919	0.05456	0.06480
Gradient Boosted Trees	0.05546	0.05308	0.05505

Table 9. Predictive Modeling Result

The best performing model was a Random Forest Regressor using the top 1000 bigrams as features for predicting which yielded a mean absolute error of around 5%. Since stock prices do not typically move by 5% within a day (even during times of high volatility such as after an earning call), our model's usefulness is limited.

We propose several reasons for why this might be the case, the first of which is our largest limitation, the small dataset. This dataset only consisted of 188 call transcripts, from 10 companies spanning a period of 4-5 years. The lack of data severely limited our options for modeling choices in terms of exploring more complex models. If we had a significantly larger dataset, other models we would have considered include deep learning models like GRUs, LSTMs, and their bidirectional variants. These models might be better

suited to capturing the sentiment expressed in those earning calls as they allow for modeling more complex relationships.

Secondly, the 10 stocks in this dataset consist of 10 popular technology companies such as Apple and Amazon. These stocks have huge market capitalization and are widely researched and covered by analysts. With this much attention, it is likely that most of the information in the earning call would have already been forecasted by investors prior to the call happening. Therefore, the sentiments conveyed in the earning calls would have already been reflected in the price even before the earning call takes place. In this case, there is little new information to be gained from the earning calls that could be used as signals to forecast the price movements.

Lastly, our model only considers the earning call text as the sole predictor of stock returns. In reality, the stock market is a highly complex system with a multitude of factors that could influence the price of a stock at any given time. Therefore, our models do not capture those influences and the possible interactions between them. Again, a larger data set of transcripts would be ideal for doing such modeling.

6. CONCLUSION

6.1 SENTIMENT ANALYSIS

Domain-specific sentiment dictionaries (e.g. Loughran-McDonald and Henry dictionaries) are shown to return more positive sentiment categorizations in general. However, there are as well more ups and downs in the overall sentiment generated by those dictionaries over time, at least from 2016 to 2020. Those all make sense since companies usually try to promote their products, idealize their strategies, expose any new

technologies, or basically PR things that could boost their stock prices through earnings call conferences.

Specifically, the gaussian-smoothed Loughran-McDonald sentiment comparison plot shows that Google had the most positive sentiment in their transcripts than Amazon or Apple did, but all three companies' overall sentiment dropped notably when the COVID-19 pandemic arose. Future work might focus on the effects of this kind of positivity based on this empirical finding.

6.2 TOPIC MODELING

Algorithm	Coherence score	K specification required	Topics generated	Topics assigned
Manual	None	Yes	8	6
LDA	0.3262	Yes	10	10
Ensemble LDA	0.2195	No	1	1
LSI	0.439	Yes	5	4
HDP	0.6394	No	20	5

Table 10. Key Features of Topic Modeling Methods/Algorithms

The table above compares the five methods we put into use. The manual method, eLDA, and HDP could be discarded because 1) for the manual method, one word is never a good summarization of a topic, and the choice of words is quite random; 2) eLDA does not partition our documents; 3) Too many topics are unused based on what HDP returns, and those topics in use are not informative enough.

However, there is no rule of thumb either to select the better one from LDA and LSI. LDA does better in resulting in more topics and each of them focuses on more detailed aspects of the whole earning call corpus, and there are more relevant auxiliary APIs to help interpret the ordinary LDA's result. On the other hand, LSI allocates the topic

in a more generalized manner, where each topic discusses different parts of the value chain from different businesses, including the customer side, product end, R&D/technology end, and fiscal concerns. Moreover, HSI returns a higher coherence score.

6.3 SUPERVISED LEARNING

We fitted several machine learning regression models in our attempts to forecast the 1-day stock return immediately following an earnings call. The best performing model was the Random Forest regressor with 1000 estimators and used bigrams of the vectorized text from transcripts. This model had a mean absolute error of approximately 5% which severely limits its usefulness in any reasonable investment strategy.

This could likely be due to several factors. Perhaps the largest limitation in our approach was the size of the dataset which limited the complexity of models we could fit. Secondly, forecasting stock prices is a tricky task and it is therefore not surprising that a simple model using vectorized earning call transcripts alone does not have much predictive power. Lastly, it could simply be the case that the earning calls in this small dataset do not happen to contain much surprise news. Since all information in the earning call is expected, it would have already been priced in and there is no predictive power to be gained from using the transcripts as features.

REFERENCES

1) Huang, A. H., Yang, Y., Oh, H., Bae, J., Chiu, P.-C., Frankel, R., Chen, K., Chen, J., & Shrestha, Y. R. (2017, June 13). Analyst information discovery and interpretation roles: A topic modeling approach. *Management Science*. https://pubsonline.informs.org/doi/10.1287/mnsc.2017.2751
2) Kapadia, S. (2021, December 11). Evaluate Topic Models: Latent Dirichlet Allocation (LDA). *Medium*. https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0
3) Labs, D. D. (2020, July 1). Earnings call sentiment analysis (part I). *Medium*. https://medium.com/@deephavendatalabs/earnings-call-sentiment-analysis-part-i-e3e7aafe2cab
4) Loughran, T. I. M., & McDonald, B. I. L. L. (2011). When is a liability not a liability? textual analysis, dictionaries, and 10-KS. *The Journal of Finance*, *66*(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x
5) Ma, Z., Bang, G., Wang, C., Liu, X. (August 2020). Towards earnings call and stock price movement. *Research Gate*. https://www.researchgate.net/publication/344084464_Towards_Earnings_Call_and_Stock_Price_Movement
6) Medya, S., Rasoolinejad, M., Yang, Y., & Uzzi, B. (2022, January 31). An exploratory study of stock price movements from earnings calls. *arXiv.org*. https://arxiv.org/abs/2203.12460

7) Nair, A. (2022, March 5). Topic Modeling with Latent Semantic Analysis - Towards Data Science. *Medium*. https://towardsdatascience.com/topic-modeling-with-latent-semantic-analysis-58aeab6ab2f2

8) Sroka, E. C. (2021, December 14). Don’t be Afraid of Nonparametric Topic Models (Part 2: Python). *Medium*. https://towardsdatascience.com/dont-be-afraid-of-nonparametric-topic-models-part-2-python-e5666db347a

9) Pawar, A. (2022, April 9). Topic Modelling using Ensemble LDA - Abhishek Pawar. *Medium*. https://medium.com/@abhi_pawar/topic-modelling-using-ensemble-lda-71d2a78666fc

10) Rehurek, R., & Sojka, P. (2011). Gensim–python framework for vector space modeling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, *3*(2).

APPENDIX - RAW OUTPUT OF PRINTED TOPICS

LDA

(0, '0.009*"customer" + 0.008*"revenue" + 0.006*"research" + 0.005*"like" + 0.005*"ceo" + 0.004*"would" + 0.004*"next" + 0.004*"product"')

(1, '0.010*"revenue" + 0.009*"cloud" + 0.007*"customer" + 0.007*"microsoft" + 0.006*"azure" + 0.006*"currency" + 0.005*"margin" + 0.005*"data"')

(2, '0.009*"product" + 0.008*"revenue" + 0.006*"customer" + 0.006*"million" + 0.005*"data" + 0.005*"next" + 0.005*"ceo" + 0.005*"technology"')

(3, '0.014*"customer" + 0.005*"next" + 0.005*"revenue" + 0.005*"like" + 0.005*"would" + 0.005*"research" + 0.005*"demand" + 0.005*"product"')

(4, '0.010*"revenue" + 0.009*"product" + 0.006*"data" + 0.006*"million" + 0.006*"cloud" + 0.005*"customer" + 0.005*"margin" + 0.005*"like"')

(5, '0.009*"revenue" + 0.008*"customer" + 0.008*"cisco" + 0.007*"system" + 0.005*"like" + 0.005*"data" + 0.005*"product" + 0.004*"cloud"')

(6, '0.009*"customer" + 0.008*"euv" + 0.007*"would" + 0.006*"asml" + 0.006*"product" + 0.006*"revenue" + 0.006*"research" + 0.006*"margin"')

(7, '0.009*"revenue" + 0.007*"product" + 0.006*"customer" + 0.006*"like" + 0.006*"cloud" + 0.005*"youtube" + 0.004*"research" + 0.004*"call"')

(8, '0.007*"customer" + 0.007*"revenue" + 0.006*"amazon" + 0.005*"product" + 0.005*"like" + 0.005*"margin" + 0.005*"research" + 0.005*"prime"')

(9, '0.010*"revenue" + 0.008*"apple" + 0.006*"customer" + 0.006*"iphone" + 0.005*"product" + 0.004*"research" + 0.004*"like" + 0.004*"margin"')

Ensemble LDA

(0, '0.008*"revenue" + 0.007*"customer" + 0.006*"product" + 0.005*"like" + 0.005*"research" + 0.004*"margin" + 0.004*"next" + 0.004*"cloud" + 0.004*"would" + 0.004*"million" + 0.004*"data" + 0.004*"ceo" + 0.003*"call" + 0.003*"analyst" + 0.003*"expect" + 0.003*"second" + 0.003*"technology" + 0.003*"result" + 0.003*"bit" + 0.003*"strong"')

LSI

(0, '-0.250*"revenue" + -0.223*"customer" + -0.178*"product" + -0.138*"margin" + -0.136*"research" + -0.134*"like" + -0.128*"next" + -0.127*"cloud" + -0.122*"would" + -0.119*"million" + -0.116*"ceo" + -0.113*"data" + -0.103*"call" + -0.103*"expect" + -0.102*"technology" + -0.101*"analyst" + -0.098*"second" + -0.094*"bit" + -0.093*"demand" + -0.092*"gross"')

(1, '-0.309*"euv" + -0.269*"asml" + 0.207*"cloud" + -0.191*"holding" + -0.178*"peter" + -0.174*"eur" + -0.173*"customer" + -0.153*"dram" + -0.137*"memory" + -0.132*"management" + -0.130*"wennink" + 0.124*"microsoft" + 0.121*"revenue" + -0.116*"board" + 0.115*"data" + 0.108*"azure" + -0.108*"tool" + 0.103*"corporation" + -0.102*"system" + -0.101*"president"')

(2, '0.232*"customer" + -0.197*"technology" + -0.170*"micron" + -0.168*"product" + 0.161*"euv" + 0.159*"microsoft" + 0.154*"cloud" + 0.150*"azure" + -0.144*"lisa" + 0.143*"asml" + 0.139*"currency" + 0.130*"constant" + -0.128*"nand" + -0.121*"micro" + -0.113*"advanced" + -0.109*"director" + -0.106*"fiscal" + 0.104*"commercial" + -0.102*"president" + -0.102*"graphic"')

(3, '-0.249*"micron" + -0.220*"technology" + -0.202*"nand" + 0.199*"nvidia" + -0.191*"dram" + -0.186*"fiscal" + 0.137*"gpu" + 0.133*"president" + 0.132*"center" + 0.131*"data" + 0.125*"learning" + -0.122*"cost" + 0.120*"deep" + -0.115*"demand" + 0.114*"system" + 0.112*"gaming" + 0.112*"computing" + 0.110*"gpus" + -0.102*"revenue" + -0.101*"bit"')

(4, '0.199*"lisa" + 0.171*"micro" + 0.171*"advanced" + 0.170*"device" + 0.164*"revenue" + 0.157*"microsoft" + -0.141*"nvidia" + 0.137*"azure" + 0.130*"margin" + 0.125*"ryzen" + -0.119*"micron" + -0.118*"learning" + -0.115*"technology" + 0.113*"independent" + 0.107*"gross" + 0.107*"constant" + 0.106*"commercial" + 0.104*"currency" + 0.103*"custom" + -0.103*"world"')

HDP

(0, '0.008*revenue + 0.007*cloud + 0.005*product + 0.005*data + 0.005*customer + 0.005*like + 0.004*next + 0.004*million + 0.004*margin + 0.003*research + 0.003*corporation + 0.003*microsoft + 0.003*call + 0.003*platform + 0.003*operating')

(1, '0.008*revenue + 0.008*customer + 0.006*product + 0.005*research + 0.004*like + 0.004*margin + 0.004*would + 0.004*ceo + 0.004*next + 0.004*second + 0.003*apple + 0.003*analyst + 0.003*nanometer + 0.003*call + 0.003*million')

(2, '0.008*customer + 0.007*revenue + 0.007*technology + 0.006*product + 0.006*micron + 0.005*demand + 0.005*dram + 0.005*nand + 0.005*bit + 0.004*data + 0.004*cost + 0.004*margin + 0.004*like + 0.004*cisco + 0.004*fiscal')

(3, '0.009*customer + 0.006*amazon + 0.005*revenue + 0.005*prime + 0.004*lot + 0.004*research + 0.004*would + 0.004*cloud + 0.004*next + 0.004*like + 0.003*product + 0.003*result + 0.003*brian + 0.003*call + 0.003*cfo')

(4, '0.007*product + 0.006*customer + 0.005*revenue + 0.005*technology + 0.004*like + 0.004*ceo + 0.004*analyst + 0.004*cisco + 0.004*micron + 0.004*margin + 0.004*system + 0.004*bit + 0.003*director + 0.003*research + 0.003*company')

(5, '0.007*revenue + 0.005*cloud + 0.005*product + 0.004*second + 0.004*customer + 0.004*data + 0.004*research + 0.003*margin + 0.003*half + 0.003*million + 0.003*center + 0.003*non + 0.003*ceo + 0.003*platform + 0.003*strong')

(6, '0.006*technology + 0.006*revenue + 0.006*customer + 0.006*product + 0.005*margin + 0.004*research + 0.004*nand + 0.004*ceo + 0.004*bit + 0.004*gross + 0.004*micron + 0.004*dram + 0.004*would + 0.004*fiscal + 0.004*analyst')

(7, '0.006*revenue + 0.005*product + 0.004*data + 0.004*million + 0.004*center + 0.004*customer + 0.003*research + 0.003*lisa + 0.003*margin + 0.003*call + 0.003*ceo + 0.003*graphic + 0.003*second + 0.003*president + 0.003*would')

(8, '0.006*customer + 0.005*revenue + 0.005*apple + 0.003*would + 0.003*iphone + 0.003*euv + 0.003*product + 0.003*margin + 0.003*system + 0.003*research + 0.003*like + 0.003*ceo + 0.003*analyst + 0.003*asml + 0.003*call')

(9, '0.008*product + 0.007*revenue + 0.005*million + 0.005*margin + 0.004*lisa + 0.004*advanced + 0.004*micro + 0.004*device + 0.004*research + 0.004*data + 0.004*customer + 0.004*next + 0.003*non + 0.003*ceo + 0.003*president')

(10, '0.006*revenue + 0.006*customer + 0.004*cloud + 0.004*like + 0.003*research + 0.003*product + 0.003*call + 0.002*next + 0.002*cisco + 0.002*investment + 0.002*come + 0.002*microsoft + 0.002*result + 0.002*strong + 0.002*search')

(11, '0.009*euv + 0.008*customer + 0.007*asml + 0.006*would + 0.005*eur + 0.005*holding + 0.005*management + 0.005*research + 0.005*peter + 0.004*board + 0.004*revenue + 0.004*system + 0.004*next + 0.004*memory + 0.004*mean')

(12, '0.007*million + 0.007*product + 0.007*revenue + 0.006*lisa + 0.005*president + 0.005*amd + 0.004*advanced + 0.004*device + 0.004*micro + 0.004*graphic + 0.004*second + 0.004*ceo + 0.004*would + 0.004*custom + 0.004*analyst')

(13, '0.006*apple + 0.004*iphone + 0.004*product + 0.004*revenue + 0.003*analyst + 0.003*next + 0.003*customer + 0.003*intel + 0.003*tim + 0.003*rate + 0.003*like + 0.003*company + 0.003*stacy + 0.002*inventory + 0.002*corporation')

(14, '0.008*euv + 0.007*asml + 0.007*eur + 0.006*customer + 0.005*memory + 0.004*would + 0.004*holding + 0.004*management + 0.004*peter + 0.004*board + 0.004*system + 0.003*research + 0.003*order + 0.003*margin + 0.003*demand')

(15, '0.008*customer + 0.007*asml + 0.006*euv + 0.006*holding + 0.006*peter + 0.005*eur + 0.005*analyst + 0.004*tool + 0.004*wennink + 0.004*nanometer + 0.004*president + 0.004*ceo + 0.004*node + 0.004*system + 0.003*million')

(16, '0.005*customer + 0.004*euv + 0.004*system + 0.003*need + 0.003*model + 0.003*asml + 0.003*gaming + 0.003*revenue + 0.003*nvidia + 0.003*like + 0.002*call + 0.002*million + 0.002*would + 0.002*holding + 0.002*president')

(17, '0.008*amazon + 0.005*com + 0.005*customer + 0.004*brian + 0.004*revenue + 0.004*prime + 0.003*line + 0.003*aws + 0.003*cfo + 0.003*margin + 0.003*olsavsky + 0.003*come + 0.003*like + 0.003*next + 0.003*cost')

(18, '0.007*prime + 0.006*customer + 0.006*amazon + 0.004*brian + 0.004*com + 0.004*aws + 0.003*olsavsky + 0.003*cfo + 0.003*lot + 0.003*next + 0.003*like + 0.003*guidance + 0.003*result + 0.002*investment + 0.002*revenue')

(19, '0.008*apple + 0.007*iphone + 0.005*revenue + 0.004*customer + 0.003*ipad + 0.003*june + 0.003*china + 0.002*product + 0.002*tim + 0.002*ago + 0.002*analyst + 0.002*watch + 0.002*channel + 0.002*qtr + 0.002*share')