# Related work (Peter, Vijay, Andrew, Harrison)

* [MetaQ] Are there any research which was not included previously in this section that you want to use to compare to the performance of something we built?
* What is the research done?
* Who did the research?
* When did the research get published?

<https://aaltodoc.aalto.fi/server/api/core/bitstreams/d7ae89d6-50c0-4b9f-b214-ab87104d2f5d/content>

<https://www.researchgate.net/publication/378435618_Stock_price_prediction_using_decision_tree_classifier_and_LSTM_network>

# Methodology

## Datasets, tools, main tasks, analytical thinking

### For each tool used

We use the python library spaCy for computing embeddings of words in a semantically meaningful way. There are several alternatives, such as Gensim, Hugging Face transformers such as BERT, GPT, or others. We chose to use spaCy because it had the simplest interface among tools for computing embeddings of words and because the memory requirements of spaCy were small compared to the other methods.

We use a k-means algorithm to compute group similar embeddings of words together into clusters. There are several clustering algorithms, such as DBScan and EM-based clustering. We chose to use a k-means algorithm over the others because the number of clusters cannot be controlled in DBScan which means that interpretation would depend on good tuning of the density parameter \( \varepsilon \) for DBScan, and because k-means algorithm outputs clusters which are more easily interpreted compared to the probability distributions which would be output by EM-based clustering.

We use an extractive summarization algorithm to remove information of low importance in SEC 10-K reports. There are several extractive summarization algorithms such as textRank, lexRank, and various BERT transformers. We chose to use textRank because that allowed for easy integration when we used spaCy for computing embeddings of words.

### For each main task

* What are the technical details of the method that we finally used?

The third main problem that we wanted to address was the poor understanding of the technology sector. To do this, we wanted to construct a more fine-grained understanding of what companies are doing within the technology sector. The primary difficulty we encountered in solving this problem was that it was unclear how to programmatically determine what a company is focusing on in the technology sector – this information is useful to competitors, so companies are motivated to not reveal this easily. To solve this problem, our initial approach was to apply keyword finding algorithms on the contents of the SEC 10-K reports. The main difficulty we encountered when applying these algorithms was that the keywords we found were semantically not important. To extract keywords which were relevant to the company directions, we used the nltk English stop word list to remove the words which appeared frequently in the text so that the keywords would be only words from the text which were semantically important (as opposed to syntactically necessary, for example ``the’’, ``a’’, ``of’’). The full details of our submitted method for this are as follows:

\begin{enumerate}

\item Read in a list of companies and associated keywords.

\item Compute an embedding of each keyword using spaCy.

\item For each company, compute the average of embeddings of keywords associated with that company.

\item For \( k = 2, \dots, 100 \), compute a \( k \)-means clustering of the keyword embeddings, and compute the inertia (for the elbow score) and silhouette score of the clustering.

\item For each cluster in each clustering, use cosine similarity to find the closest keywords in the cluster to the centroid of the cluster.

\end{enumerate}

The fourth main problem that we wanted to address was understanding what strategic directions cause company stock performance to rise or fall. To do this, we wanted to construct a simple set of rules which would classify whether a stock price is predicted to rise, or fall based on the strategic directions the company is pursuing. Our main tool used for this task was a decision tree. Our initial attempt at constructing the decision tree used a computed sentiment score and a Boolean vector for a set of keywords which encoded whether a keyword was in the set of keywords associated to an entry in the dataset (that is, we used a one-shot encoding for the keywords). The main difficulty we encountered with the initial approach was that the information gain for each of the keyword features was not large enough for the decision tree algorithm to construct any layers beyond the root node’s layer. To address this difficulty, we constructed clusters of keywords using k-means and constructed a feature for the number of keywords belonging to that cluster in the list of keywords for the datapoint. The full details of our submitted method are as follows:

\begin{enumerate}

\item We compute the set of all keywords in the dataset.

\item We then compute a \( k \)-means clustering of the keywords.

\item For each datapoint, for each keyword, we count how many of the keywords for the datapoint are in each cluster.

\item We convert the percentage change to a label (specifically, the labels correspond to if the percentage change is in the interval \( (-\infty, -3], (-3, 3), [3, \infty) \)).

\item For each company in the dataset, train the decision tree using the datapoints corresponding to all other companies.

\item Evaluate the decision tree accuracy on the remaining (non-training) datapoints.

\end{enumerate}

### Analytical thinking

For our goal of improving understanding of the technology sector, we initially expected to find major trends in technology aligning with ``buzz words’’ often used to advertise to investors to exploit the so-called fear-of-missing-out of major trends. Because SEC reports have only been available for the past 25 years at most, we expected to primarily find ``buzz words’’ related to artificial intelligence and big data. We collected data primarily from technology sector companies because we were focused on verifying whether this pattern holds. We collected data from a small number of companies not in the technology sector to ascertain the consistency of finding buzz words even in more ``typical’’ conditions where one might analyze several different market sectors simultaneously. We also collected data from companies which have recently had poor stock performance to build a more accurate decision tree (as stock prices and company performance have not become better within the whole market in recent years).

We can interpret the results from our k-means partitioning of keywords directly as a partition of companies in the technology sector. Smaller clusters of companies based on the technologies they are working on will give better explainability behind the reason some companies perform better, and some companies perform worse under market conditions. However, this analysis is not prophetic – it relies on having a trend already exist, which means we cannot use this to mechanically produce innovation; this still seems to require some human ingenuity. We can interpret the results from the decision tree construction as a set of rules predicting whether the price of a stock will increase, decrease, or stay the same based on company outlook and direction. This analysis does not include several other factors which may be important for company survival such as employee retention, or changes in leadership (and it is not clear how these factors could be encoded to be part of the decision tree).

## Data understanding, preprocessing, warehouse, modeling

### Data understanding

To understand the general conditions of the market over time, we constructed a graph which computes the sector average stock price for several different sectors. From this, we can observe that the technology sector has been growing quite significantly (more so than other sectors) in the past few years. This analysis informed us that the technology sector would be useful when we wanted to uncover key company directions and sources of success.

### Preprocessing

When working with the 10-K reports we scraped from the SEC website, we had to clean the report because the text extracted also contained meta information about the report such as page numbers. When extracting stock data, we sometimes would find missing data. The main reason for this occurring is because the stock data was requested for a period in which the company was not publicly traded. For the purposes of constructing partitions of the technology sector and constructing decision trees, we were able to ignore these entries.

We also needed to extract keywords from the 10-K reports. To do this, we applied extractive summarization on the 10-K report to extract important passages from the 10-K report. Then, we removed stopwords from the important passages and used term-frequency inverse-document-frequency (tf-idf) to identify keywords from the summary passages. We initially tried using (tf-idf) without using an extractive summarization algorithm, but this resulted in having irrelevant keywords when using tf-idf.

### Modeling

To construct a better understanding of the technology sector, we used a k-means clustering algorithm followed by an interpretation step (refer to section [TODO] actual section number here). This process takes as input a list of companies, keywords associated to their focuses, and a number of clusters~\( k \) to output. The process outputs a set of~\( k \) clusters of companies and interpretations for those clusters. We expected that this process would be effective because most companies promote their most advanced projects, products, and technologies to investors, the intended readers of the SEC 10-K reports. One major assumption we made is that the contents of the SEC 10-K reports is reflective of the things that the company submitting the report is pursuing or intends to pursue – we verified this assumption for a small number of 10-K reports, but were unable to implement a program to verify this for all the 10-K reports in the dataset. For the process, the number of clusters generated by the k-means algorithm is an important parameter as it could affect the interpretability of the output.

To better understand the effects of business decisions in the technology sector on stocks, we constructed a decision tree. The decision tree takes as features a summary of the keywords in companies 10-K reports, a quantitative sentiment score, and a number~\( k \) of clusters to compress the keyword features into. The output of the model is a decision tree which outputs a classification as to whether the price of a stock moderately increased, stayed about the same, or moderately decreased. We expected that this model would be accurate because of the common theory that there are ``market trends’’ and that these market trends are reflective of all available information in the market – more specifically, we expect that the efficient market hypothesis holds, which implies that the prices of stocks are based on the underlying valuations of technologies developed by companies. When building this model, we also had several hyper-parameters which would affect the quality of the generated model; namely, all the standard hyper-parameters in addition to the number of keyword clusters. The two hyper-parameters we adjusted in training the model were the max depth of the decision tree and the number of keyword clusters to be used for condensing the keyword feature vectors.

# Evaluation

Metrics

* What are we measuring about the model – how do we know our model is good?
* What statistics can we state about the model quality?

We used two methods, the elbow method and silhouette score, to verify the quality of the k-means clusters of companies based on their key technologies and directions. We computed the following statistics for the model quality:

We measured the quality of the decision tree using a weakened \( k \)-fold validation. Specifically, for each company, we trained the decision tree on the data of all other companies in the dataset and then computed the accuracy of the decision tree in predicting the price change on the original company. We computed the following statistics for the model quality:

Evaluation setup

* How do we train the models?

Baseline methods

For clustering companies based on key technologies and directions, we found a study which also proposed to use k-means clustering for industry classification. There are three main differences between [name’s] work and ours. The main difference between our method and theirs is that we focused primarily on cluster coherence and interpretability, whereas the prior work is focused on optimizing clusters based on similar stock price changes.

There is no prior work explicitly using keywords and sentiments as features for classification of stock price change, but there is prior work which uses past stock prices as a feature for classification. The main difference between the two studies is the number of features in the prior study is limited compared to our study. Another difference is that the number of possible outputs for our study is three (increase, decrease, or stay relatively the same), whereas the prior work only considers two (increase, decrease).

Results (figures, tables w/ proper labels)

* What input did we give to the model?
* What outputs did we get from the models?

Describe inputs with screenshot. Insert picture of clustering and picture of accuracy measure. Add a picture of clustering with label.

Describe inputs with screenshot. Insert one graph for two of the cross-validation runs, corresponding table for two specific hyperparameter settings, and the decision tree for the runs. Also add picture for average accuracy among different stocks in cross-validation. Note that accuracy is like prior work for technology companies – the main difference seems to be the quality of features.

USE ADP pictures, NVDA pictures, and AMC pictures. Also add computed clusters for better interpretation.

Interpretation

* What can we learn generally from the results we got from the model?
* Are there any specific, interesting, or surprising findings from our results?

We found that the clustering of companies was generally quite revealing of what underlying modern technology sectors there are. When the number of clusters is small, we found that we were able to obtain high-level industry trends which correspond to existing modern technologies. [Insert labelled picture with categories here].

We found generally that our methods were most effective when classifying changes for technology companies. For companies other than technology companies, we had quite poor accuracy compared to the technology companies. This seems to be because the number companies not in the technology sector were not large enough to generate clusters of keywords which would be primarily relevant to the companies not in the technology sector. When compressing the keywords into four keyword clusters, we were able to observe that the clusters approximately corresponded to the following technologies:

# Discussions

Lessons learned

A major technical challenge which we did not consider in the class is whether a cluster from the k-means clustering algorithm can be interpreted, and how it can be interpreted. While there are some algorithmic ways to interpret the clusters, we found that the interpretation of the clustering is best left for humans. This is reflective of the ideal data mining pipeline; that user interaction should be an important part of the pipeline because the goal of data mining is to mine patterns for human use cases from the data.

What worked well/what didn’t

We found that the k-means clustering was able to give very reasonable clusters for the classification of subsectors of the technology sector. This is likely because of the quality of keywords extracted. This suggests that algorithmic analysis of financial documents will need to be improved from a ground-up approach; before the document can be accurately processed and conclusions drawn from the document, the process to semantically analyze financial documents will need to be of sufficiently high quality.

We found that the accuracy of the decision tree classifiers was not very high and did not generally without significant changes in the hyperparameter values. This is because our hyperparameters were conflicting – more specifically, even if we allowed for more clusters of keywords as features, limiting the depth of the decision tree prevents the decision tree from using many of the clusters. This could be avoided by increasing the maximum depth to be greater than the number of clusters.

Directions for future work

There are several things that could improve this project. We could increase the number of stocks and market sectors studied by our method; this project focused primarily on the technology sector and only referenced a small number of companies outside of the technology sector. We could also try to integrate more types of public filings in our data set: this project primarily focused on analyzing 10-K reports which, while comprehensive, do not report any sudden changes which would be reported in other reports.

A new technique that is not explored in this paper but may be useful for the algorithmic analysis of latent trends affecting stock prices may be to draw keywords and news from media and other press releases. This would allow more data to be collected, allowing for more keywords to be identified for each company. However, confirming the veracity of the keywords and news would then need to be addressed – this seems like it would require more sophisticated techniques in artificial intelligence.