SECret Insights:
Unsupervised Findings from SEC 10-K Filings

Rahul Jain, Keyur Shah
Cornell Tech
New York, NY 10044
rj299@cornell.edu, ks957@cornell.edu

December 7, 2022

Abstract

Companies are required by law to file the 10-K form annually. This includes an overview of the firm's business, risk factors, selected financial data, management's discussion and analysis, and financial statements. We have analyzed a snapshot of this information from 2020 using unsupervised learning algorithms and discovered some hidden insights. Mainly, we have attempted to cluster the companies based on the text and tabular financial data. We have found that some clusters did reflect some intuitive real-world meaning after we performed thoughtful preprocessing and algorithm selection. However, we still didn't capture the full depth of structure that we expected, suggesting that more data would be needed to truly understand a company.

1 Introduction

The 10-K forms that companies submit contain two components that we will analyze: a written (text) document explaining the current state of the business and a (tabular) financial statement with specific numbers calculated by accountants/auditors. We aim to complete an application project, analyzing both of these datasets using unsupervised learning algorithms, comparing their usefulness for the clustering problem and then visualizing them to discover any hidden insights. We will do this by applying well-known machine learning and natural language processing techniques. The first dataset [1] contains the text portion of 10-K (annual) financial disclosures for 2020, and the second dataset [2] has the financial statement portion for a different (but overlapping) set of companies from 2008-2021. We will join this data on company name, keeping only the year 2020, and cluster the companies to see what groupings emerge, using the datasets separately and also both together.

We believe this work is interesting because we are analyzing how "real" the relationship is between the written part of financial filings and the hard numerical part. Many hobbyist and professional investors scrutinize these filings every year to develop mental profiles of companies, but they often deal with the challenge of how to put the many data points together. Our work could suggest the importance of each of these parts and perhaps which sections of the 10-K are most important for prospective investors to be drawn to if they'd like to understand a company better.

2 Background

Broadly, the reader consuming this work should be loosely familiar with two areas of study: US corporate financial filings and unsupervised machine learning (ML) techniques. Within the domain of corporate finance, cursory knowledge about the meaning of major financial metrics would be helpful. Although we don't use many of the obscure features provided in the original dataset (because they

Fall 2022 Cornell University Applied Machine Learning Final Project.

are not relevant to the subset of filing years and company IDs we study), some understanding of what the chosen features mean may make our results more useful to the reader. In particular, a key result of ours was comparing the usefulness of the text data obtained from written portions of the company filing with tabular data obtained from the financial statement. Therefore, the reader may benefit from having used these data points before to analyze a company.

Within the domain of ML and natural language processing (NLP), the reader should primarily be familiar with common NLP preprocessing methods and ML algorithms. Specifically, knowledge of the unsupervised task of clustering would be important as we used multiple clustering algorithms with different mathematical properties and some more advanced metrics to compare them. Finally, common python packages such as scikit-learn and nltk are used to implement the algorithms so an understanding of this would be helpful if the reader wishes to follow the code.

3 Method

The first step of our project was data cleaning and preprocessing. Even though our dataset consists of highly regulated financial data, the data was not as complete as we had hoped. To even choose our dataset, we combed through multiple Kaggle archives and found the ones that seemed to be the most comprehensive and trustworthy. After this, the first challenge was that the tabular and text datasets did not share the same sets of companies. We dealt with this by only analyzing the intersection of both sets, which was still 104 companies' combined filings from 2020, which we felt should still be an interesting dataset.

Next, the text preprocessing steps that we performed included transforming to lowercase, replacing special characters, HTML tags, & escape characters (newline, tab), replacing contractions, stripping punctuation, removing stop words, numbers and finally, lemmatizing the text. The tabular dataset preprocessing included selecting attributes that were filled out for a high percentage of companies and using sklearn's StandardScalar to rescale the data. We chose the threshold by plotting the histograms for the distribution of how many fields were present across 10-K forms for companies. After manual inspection of these, we were reasonably happy that they captured many important values, including "Assets", "LiabilitiesCurrent", "StockholdersEquity", etc. We imputed the few remaining missing values with column medians. Then the values were scaled with StandardScalar, and we performed an exponential transform using the function $f(x) = 1 - exp(-\alpha x)$ to get a better distribution across the tabular data values. We also tried logarithmic scaling, where we would shift all the features to be strictly greater than 0 and then took the logarithm, but the distribution seemed skewed in the other direction, and the clustering results were poor. Thus, we decided to use the exponential transform.

Originally, we converted the text data to numerical vectors using simple Bag-of-Words (BOW) with a few simple corrections. In the final analysis, we used the TfidfVectorizer in Sklearn, which is commonly used to vectorize text by taking into account the frequency of words in the corpus. This is different from BoW, which looks at the presence of words. To keep the dimensionality under control, we tested various thresholds for how many features (i.e. words) to retain. Then we concatenated all text fields together rather than treating them as separate features with separate vocabularies. Finally, we used the unique company identifier (CIK) values to join the text and tabular data.

For our analysis, we applied a few unsupervised clustering methods to both the text and tabular data separately. We took the TF-IDF vectors for the text dataframe, and the rescaled numerical values in the tabular dataframe. We tried a variety of K values and empirically determined the best K using the elbow method for K-means. For this clustering, we then extracted the company names placed into one cluster to see whether we could identify any human-interpretable relationships. In addition to K-means, we tried using Gaussian Mixture Models (GMMs) and Agglomerative Clustering to cluster the tabular and text data for which we used the same number of classes K determined from the elbow method on K-means.

Once we had the cluster assignments for the companies using both the tabular and text data applied to the three clustering algorithms, we could attempt to generate some metrics. We used sklearn's silhouette_score, adjusted_rand_score, and adjusted_mutual_info_score to compare how close the clustering methods were and also to determine which clustering method would be good for further analysis. The silhouette_score is computed as the mean intra-cluster distance, and the mean nearest-cluster distance for each sample which represents how dense and well-separated the clusters are. The adjusted_rand_score computes the correctly classified pairs of items and is primarily used to compare

one classification clustering to another classification clustering. The adjusted_mutual_info_score measures the agreement between two clustering assignments and is normally used to compare ground truth labels to predicted labels. In our case, we used this metric to compare the cluster assignments produced by text, tabular, and both datasets since ground truth labels were not available. We also considered other unsupervised clustering metrics from the sklearn documentation but ultimately chose these as they were the simplest to interpret [3].

After making comparisons across the clustering assignments, we attempted to visualize the clustering in 2 dimensions using dimensionality reduction techniques such as Principal Component Analysis (PCA) and T-distributed Stochastic Neighbor Embedding (t-SNE). From here, we could match up labels from clustering using text and clustering using tabular data to plotted points to see what clusters emerged.

4 Experimental Analysis

We believe that much of our time and ingenuity went into data preprocessing. On the financial (tabular) data, we were able to take data in a somewhat unfriendly structure and extract some important fields automatically. However, in our initial attempt, our scaling method produced highly skewed feature distributions dominated by large outliers, and we believe it led to unbalanced pairwise distances when computing similarity during the later clustering step. The output of clustering on this data was clusters of wildly different sizes, so improving this was a key priority for us. We achieved this improvement by finding a good transform of the input data to remove the skewness. As described above, although the data histogram at first suggested a log-transform, we found that a more complicated $f(x) = 1 - exp(-\alpha x)$ transform (where we tuned $\alpha = 3$) produced better results empirically. We fully expected highly skewed data as our real-world knowledge of public companies suggested that they do have a skewed distribution.

For the text data, our cleanup steps (such as eliminating stop words and splitting contractions) removed some of the non-useful tokens in the text. Our first attempt at vectorizing this variable-length string into a usable array was the simplest possible method, binary Bag-of-Words. However, we didn't believe that this was sophisticated enough since, for example, a standard list of stop words wouldn't catch much of the repeated boilerplate language that would probably be part of a legal report. To mitigate this, we used TF-IDF which would discount the importance of these corporate-finance-specific stop words while promoting the words that were highly specific to a company.

Once we applied all this preprocessing and experimented with different clustering algorithms, we observed using the elbow method on K-Means that the ideal number of clusters seemed to be roughly 10 (see Figure 2 in Appendix). To allow an apples-to-apples comparison between the algorithms, we fixed this choice for the number of clusters and reran clustering using each. Since this is an unsupervised learning project, we did not have ground truth labels, but we could compare the results of different algorithms and different datasets (tabular vs. text vs. both) against each other. When we evaluated these different clusterings using the clustering performance evaluation metrics described above, we got the results described in Tables 1, 2, 3 in the Appendix.

As seen in the tables, the effect of algorithm choice wasn't very strong. From the results of the silhouette scores, GMM seems to yield the best clustering result for tabular data (score of 0.362), but Agglomerative Clustering gave slightly better clustering results for the text and combined data (scores of 0.141 and 0.276, respectively). Additionally, the adjusted rand scores and adjusted mutual info scores (for all algorithms but especially Agglomerative Clustering) seemed to imply that tabular data contributed more to forming strong clusters than text did. This is because even when given both sets of data, our clustering algorithm still tended to use the tabular features and produce clusters similar to the tabular-only clusters. For this, the highest rand and mutual info score across all settings (0.837 and 0.908, respectively) came from the "both vs tabular" with Agglomerative Clustering. Furthermore, the scores for the "both vs tabular" column were higher than the scores for all other columns, which further validates our hypothesis. Our conclusion from this was the same as what we have discussed many times in class- that often times, all models find roughly the same meaning and better data is much more important than better models. The most important takeaway from these metrics is that for all scores, the "both vs tabular" clusterings were very similar, whereas "tabular" was very different than "text". We were also expecting the combination of both feature sets to be

much stronger than each dataset on its own, but instead, it seems like the text data didn't provide much extra useful information. This was clearly true regardless of algorithm choice.

Finally, we wanted to visualize the clustering to make some judgment of whether the clusters produced were "good" and reflected the real-world meaning we would expect about the companies. To create this visualization, we tested linear dimensionality reduction (PCA) and non-linear dimensionality reduction (t-SNE) and plotted both clusterings as two different properties on each, selecting PCA as we thought it showed better cluster separation and cohesion. We chose to analyze K-means on PCA in more detail as the metrics showed that the choice of algorithm did not have a heavy influence on the cluster assignments. The plot for clustering using K-means on PCA is shown in Figure 1 below. The appendix contains a breakdown of the results of applying different clustering algorithms with PCA and t-SNE (Figure 3) as well as all the companies labeled for K-means on PCA (Figure 4). For the plot shown in Figure 1, the clusters were not what we expected. Whereas we primarily thought of companies by industry or product type, the clusters that emerged did not seem to reflect similarity of industry very much. They occasionally did, with for example several audio/sound companies grouping together in one cluster, and some enterprise developer software clustering together in another, but in general, company types like investment managers and cloud security providers were spread across multiple different clusters. The one cluster that did match our prediction was the brown one in the bottom right, which grouped many big tech companies together. However, we suspect that they were actually grouped by their large size, not by similarity of product. This makes sense because we know from above that the financial statements were mostly responsible for cluster assignments, not text data, and therefore companies with similar financial numbers (i.e. sign and magnitude) would be cast together.

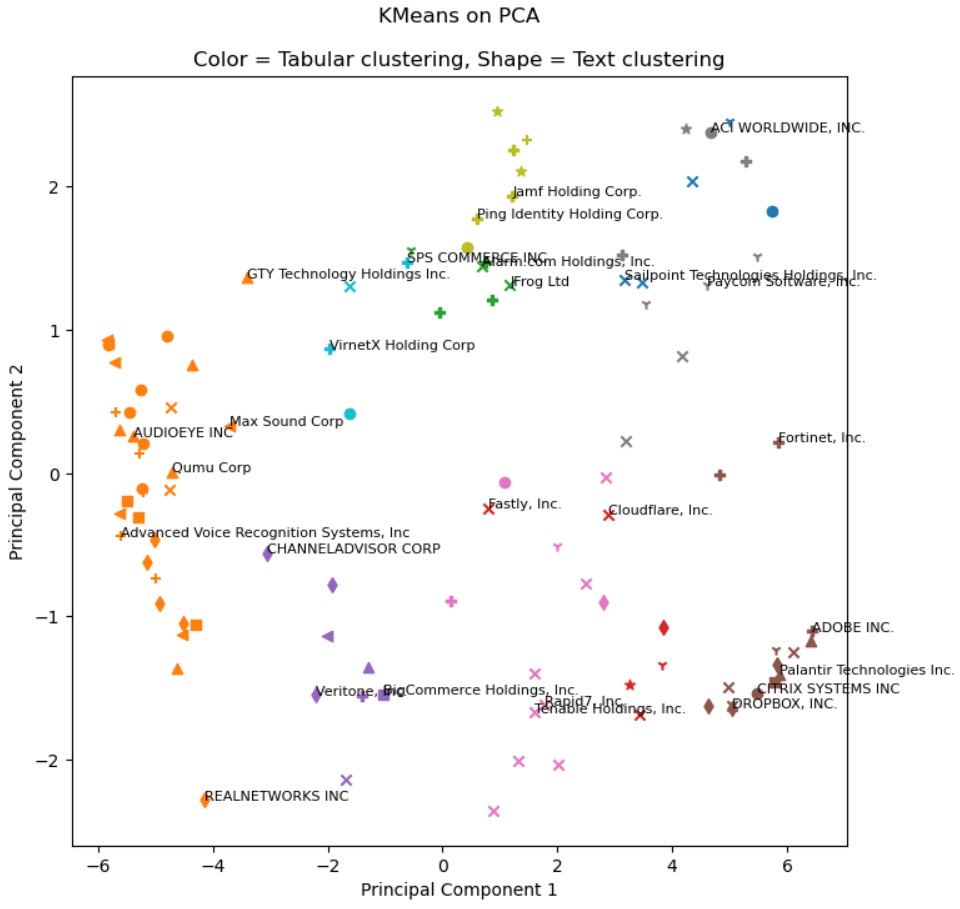

Figure 1: K-Means on PCA with select companies labeled

5 Discussion and Prior Work

Our experiments did help us make some sense of the messy data we were given, although it was not what we expected beforehand. We were very surprised that the text data was less useful to the clustering algorithms than the financial data. This could be because a 10-K filing is not the best text representation of a company–it is only filed once per year and with a very specific target audience and goal in mind. For example, while "the numbers don't lie", every company would want to position itself in the subjective writing portion as a promising enterprise with a bright future. This may have led to less variance in the vectorized text data that could be used to distinguish between clusters.

Financial data seemed to be more useful to the clustering algorithms, and the interesting output of our tabular clustering helped us realize the flaws in our original hypotheses. The way humans describe and compare companies is usually based on their industry, product, or size. For example, we think of Apple as a massive tech company that makes hardware. However, the "training data" that informs these representations of companies is very different than in this project–we get to use the company's products, whereas we don't get to see their financial performance. From a human's perspective, a massive tech company making huge profits and another making huge losses *feel* very similar, but their financial statements would look completely different. Furthermore, the clusters that emerged from the financial statements are likely to say more about financial performance, such as cash flows or assets and liabilities. To create the structure we expected, we probably would need to augment our dataset with additional features which capture that.

As a broad takeaway, we realized that deeply analyzing this dataset would require a lot of domain knowledge about this specific application domain of corporate finance or investing. A seasoned investor may have identified hidden structure that we missed in our clusters, based on better memory and intuition about the financial performance of the companies in the dataset. We believe that while our analysis makes a new contribution above and beyond other published findings, which do not compare the importance of text versus tabular data, we could still make additional improvements by procuring additional data and consulting domain experts.

6 Conclusion

In summary, we have successfully analyzed both text and tabular data for company 10-K financial forms. We have performed some pre-preprocessing on this dataset, featurized the input attributes and applied unsupervised clustering algorithms to see if we could extract any trends. Among the clustering algorithms we considered (K-Means, GMM, and Agglomerative clustering), we used some unsupervised clustering metrics (silhouette, adjusted rand, and adjusted mutual info scores) to inform our decision as to which clustering algorithm we should analyze in further detail. Before performing the analysis, we expected the text data to be more useful for clustering, but the results showed that the tabular data was more useful. Overall, there is room for potential improvements through different preprocessing and unsupervised learning algorithms, but we have shown that there is likely some real-world meaning behind what the algorithms are able to output.

References

[1] Verma, P. (2022, April 14). SEC Edgar Annual Financial Filings - 2021. Retrieved November 8, 2022, from https://www.kaggle.com/datasets/pranjalverma08/sec-edgar-annual-financial-filings-2021

[2] Chartma. (2022). SEC company facts - all 10-Q & 10-K financial data. Retrieved November 8, 2022, from https://doi.org/10.34740/KAGGLE/DS/2292262

[3] Scikit-learn developers. (2022). Clustering Performance Evaluation. Retrieved December 6, 2022, from https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation

A Appendix

	Text	Tabular	Both
K-means	0.13796376573984961	0.3084673194185171	0.2344607528161631
GMM	0.10778857484950198	0.36208922387461634	0.2254135608473509
Agglomerative	0.14060119389023282	0.2784350157850304	0.27603835532534793

Table 1: Comparison of silhouette scores across clustering methods and datasets

	Tabular vs Text	Both vs Text	Both vs Tabular
K-means	0.0969207482016067	0.09265679079548977	0.6701090003712724
GMM	0.048493401716484134	0.051401906703504495	0.595946742145022
Agglomerative	0.03493210643919559	0.03325278090354	0.8371649021519658

Table 2: Comparison of adjusted rand scores across clustering methods and datasets

	Tabular vs Text	Both vs Text	Both vs Tabular
K-means	0.16970660918638528	0.17300801677711164	0.7702568868134696
GMM	0.07847559653937938	0.08861351261418977	0.6795123372222107
Agglomerative	0.10111159141338752	0.10835764296876374	0.9087763994185828

Table 3: Comparison of adjusted mutual info scores across clustering methods and datasets

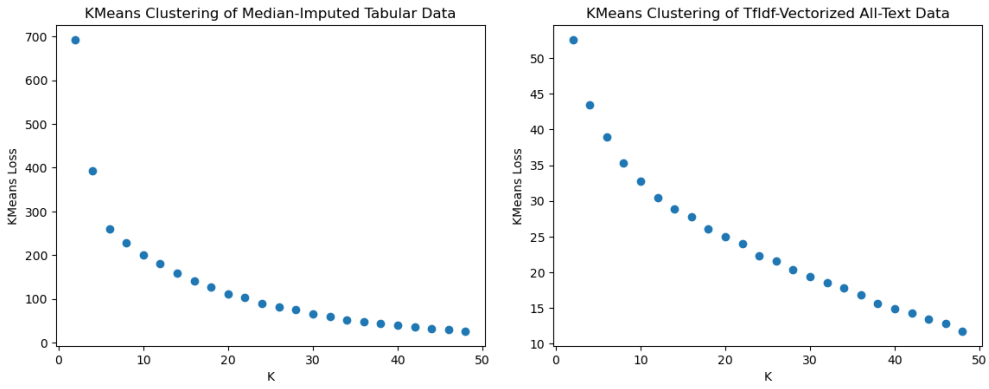

Figure 2: Elbow method performed on Tabular (left) and Text (right) Data

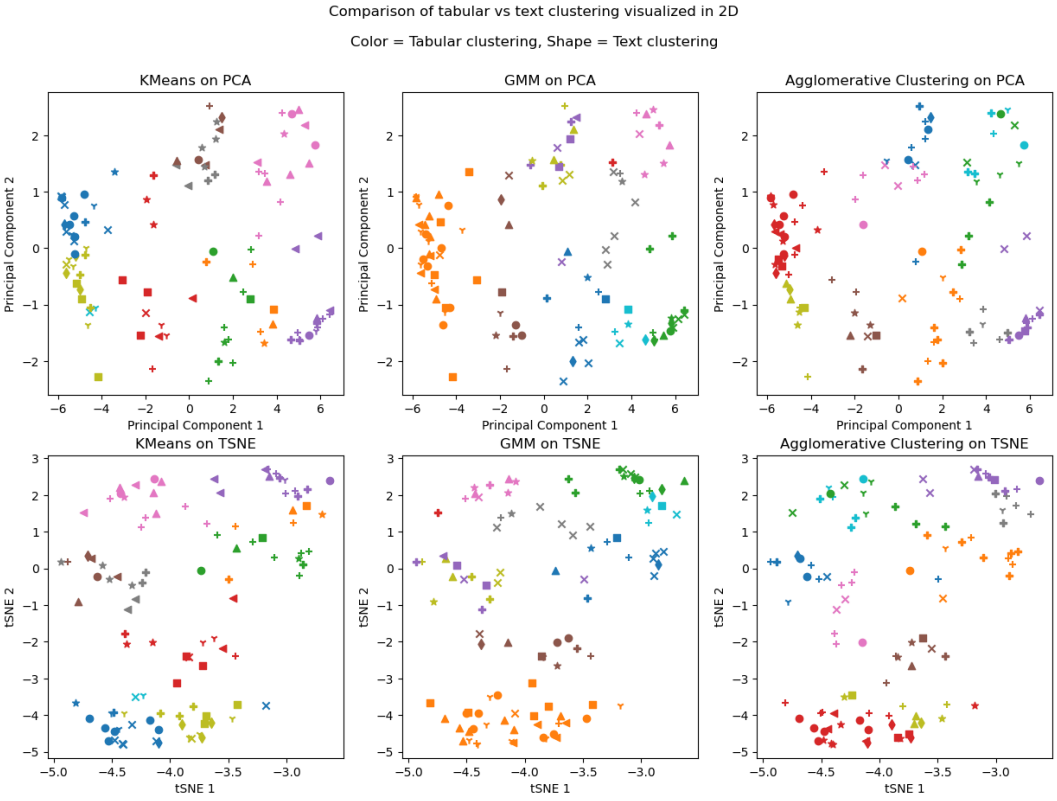

Figure 3: Comparison of tabular and text clustering methods

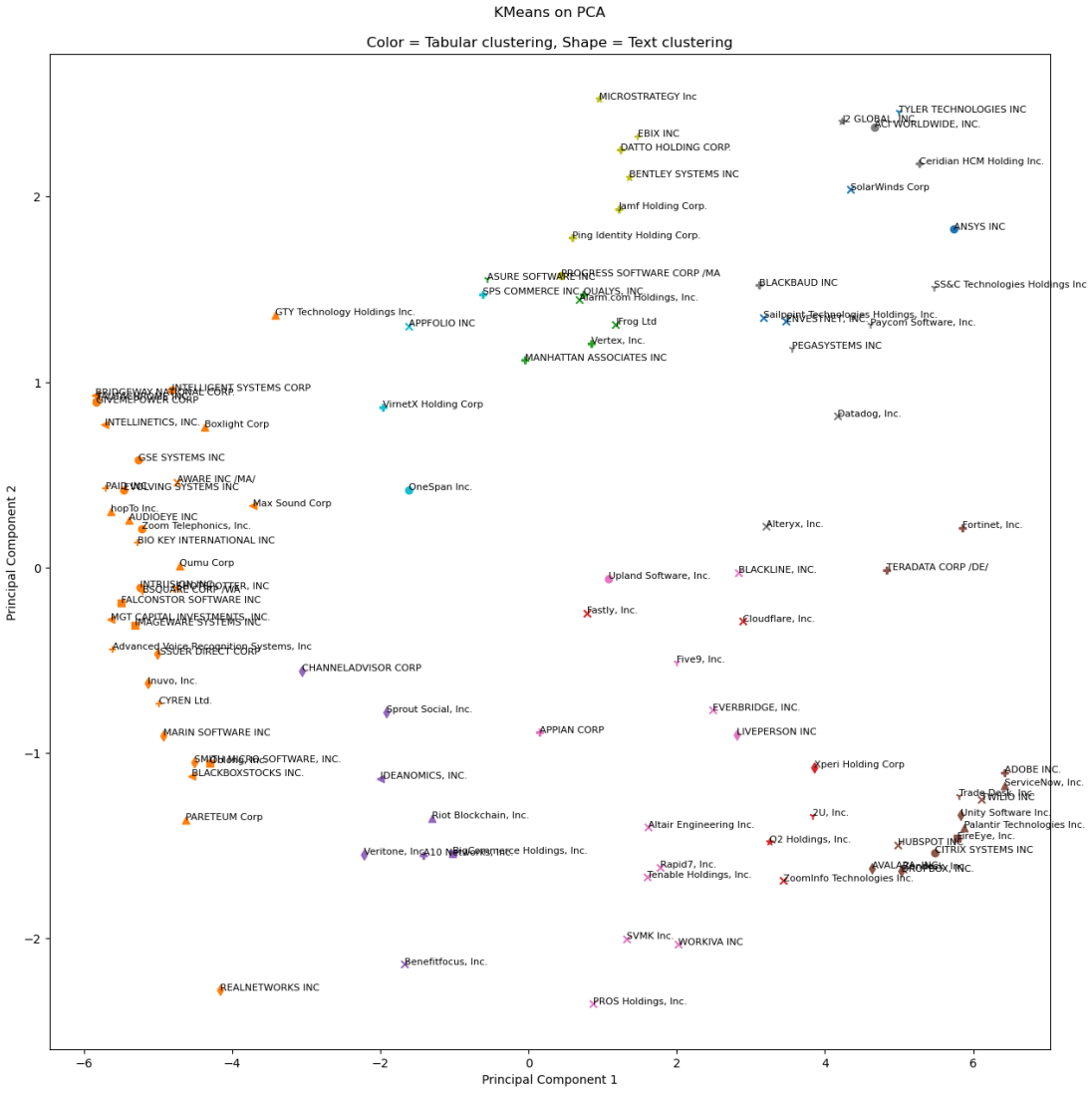

Figure 4: K-Means on PCA with all companies labeled