To perform LDA, we need to preprocess the document by Tokenization, removing the stop words, stemming and lemmatization. Here We use the NLTK and gensim libraries to perform the preprocessing.

Then we convert text to bag of words and create a dictionary reporting how many words and how many times those words appear.

Then LDA is ready to be applied, I used randomly selected data as training data, by tuning an appropriate number for topics and then applied them to different quarters in 2013 and 2014.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2013 | | | | 2014 | | | |
| Q1 | Q2 | Q3 | Q4 | Q1 | Q2 | Q3 | Q4 |
| Topic | Tech | Lifestyle | Retail | Finance | Politics | Healthcare | Lifestyle | Energy |
| Strategy | Retail | Tech | Retail | Finance | Tech | Finance | Healthcare |
| Healthcare | Politics | Finance | Tech | Tech | Finance | Tech | Finance |

Below are the findings for the year 2013 and 2014. Among which frequently brought up topics are Finance and Technology.

For part 2, I selected 2014 Q2 as my material, on the basis of removing very rare words and very common words, I selected 500 tokens and applied to every article in 2014 Q2 to generate a bundle of features of tf-idf for the tokens. Another part is LDA-based features, like TF-IDF, create a matrix of topic weighting, with documents as rows and topics as columns to show the underlying probability.

As there are too many features, we need to perform PCA to reduce the dimension, before that, labels are created by clustering. By using the k-means method and silhouette after scaling the data, I chose 4 as the best number for cluster and labeled all of them.

After adding the label, we are now able to perform the PCA, standardizing the data beforehand, I found the best number of PCs is 2, and then performed PCA to 2D, and plot the projection eventually. All the clusters can be seen from the plot.

