Natural Language Processing for Company Business Descriptions: TF-IDF Analysis and Updated Methods

Wei Ding, Wusheng Liu, Kwok Ping Ng
Technical University Munich
wei.ding@tum.de, kwokping.ng@tum.de, wusheng.liu@tum.de

1 Neural Network Classifier

In addition to the machine learning classification methods like "SVM" or "K-Nearest Neighbour", we also tried the deep learning neural network to do the category classification for the companies.

A neural network is a deep learning technique for prediction. It is built by multiple nodes and several hidden layers to stimulate a model for data to predict some results like category of a website in our case. According to universal approximation theorem, a feed forward neural network can approximate any functions or models with appropriate activation functions in nodes and parameters.

We split the data into 60% training, 20% validation and 20% testing. We use TF-IDF vectores for all seven categories from training data as the training input data for our simple neural network model. We tried different number of layers with different number of nodes in each layer. We also tried different activation function and different optimizer with some optimization tricks like weight decay. The best accuracy from the neural newwork model is a model with 3 hidden layers(Size of each layer is 4096,1024 and 256), using ReLu activation function and Adam optimizer. However, the best prediction accuracy is 56% which is still lower than the results from SVM classifier. So we didn't explore more about the neural network method.

2 New Website Vectorization Techniques

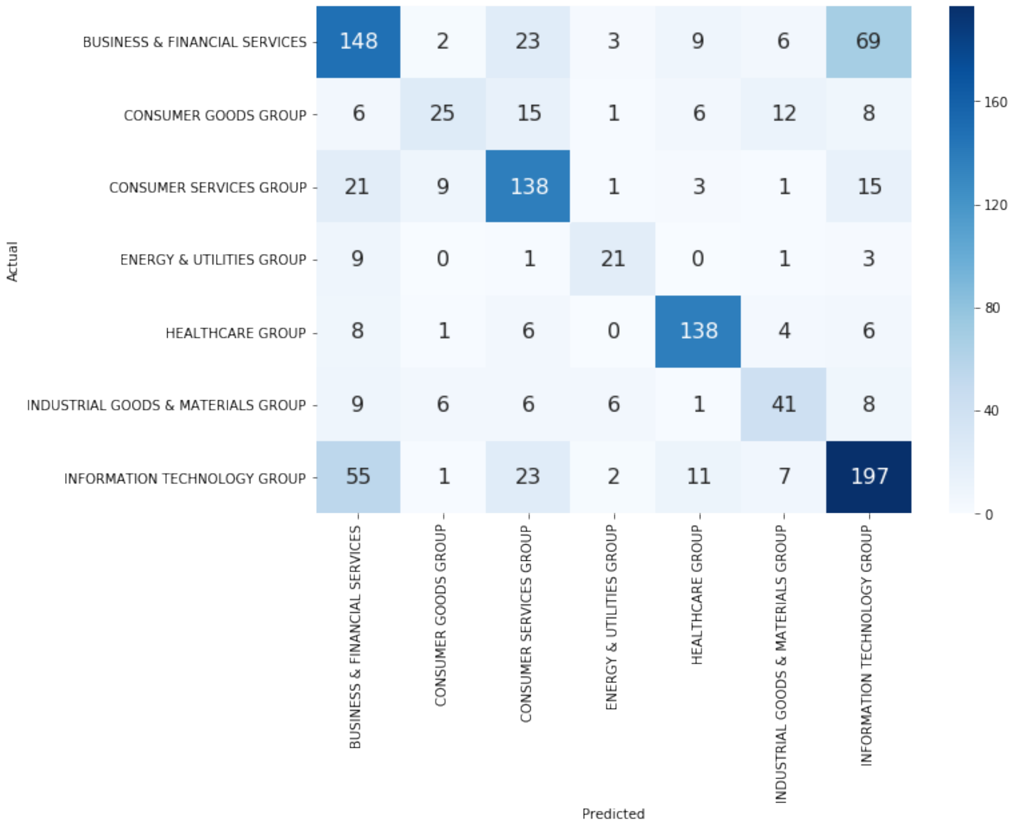
Beside the TF-IDF method to get vectors for different website, in the last two weeks, we have also explored other two techniques to generate vectors: one is using the Word2Vec model based on the results of TF-IDF. The other is using the Doc2Vec model.

2.1 TF-IDF with Word2Vec

Word2Vec[Mikolov et al., 2013] model uses to generate vector embedding for each word. This model was developed by Tomas Mikolov in 2013 at Google. It is claimed that the embedding is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc.

However, Word2Vec model only generates vectors for each word. In order to use this model, we first analyse the result of TF-IDF model. We chose N words with the highest TF-IDF value in each website. Then we generated embedding for these N words using Word2Vec and calculated the average of these vectors to get final vector for each company. The size of vectors for each website is 300. After that, we put the vectors into SVM classifier(80% training data and 20%testing data) and get the classification result.

For choosing the best N, we tried 10, 50, 100 and 200. Our model to get vectors using **100** top TF-IDF score words achieves the accuray **65%**, which is the best so far in our all experiment. Figure 1 shows



Actual \ Predicted	BUSINESS & FINANCIAL SERVICES	CONSUMER GOODS GROUP	CONSUMER SERVICES GROUP	ENERGY & UTILITIES GROUP	HEALTHCARE GROUP	INDUSTRIAL GOODS & MATERIALS GROUP	INFORMATION TECHNOLOGY GROUP
BUSINESS & FINANCIAL SERVICES	148	2	23	3	9	6	69
CONSUMER GOODS GROUP	6	25	15	1	6	12	8
CONSUMER SERVICES GROUP	21	9	138	1	3	1	15
ENERGY & UTILITIES GROUP	9	0	1	21	0	1	3
HEALTHCARE GROUP	8	1	6	0	138	4	6
INDUSTRIAL GOODS & MATERIALS GROUP	9	6	6	6	1	41	8
INFORMATION TECHNOLOGY GROUP	55	1	23	2	11	7	197

Figure 1: Confusion matrix of the classification accuracy using TF-IDF with Word2Vec model

the confusion matrix of this classification result. We may notice that this model classify the most cases correctly.

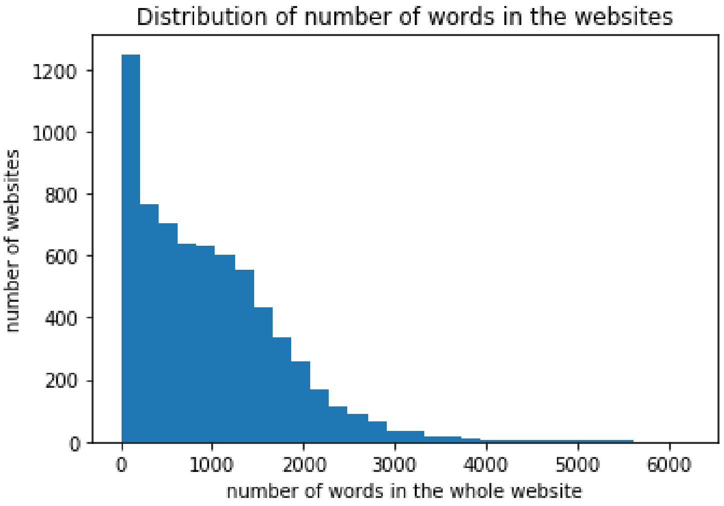


Figure 2: Historgram of word count in each website

We think the main reason for this good result is that we filtered all the words in the website to find the most representative words. Also Word2Vec is a powerful model to generate vectors for these words. Besides. we also involved a an additional data cleaning step in this model. Figure 2 shows the histogram of word count in each website. x-axis labels the number of words in each website, and y-axis means the number of companies. We can see that more than 1000 companies have less than 200 words in the whole website. So we removed these websites with words less than 200. Figure 3 shows the number of websites in each category with the category percentage in the whole corpus.

We randomly selected 300 vectors of the website generated by the TF-IDF with Word2Vec model and plotted them into two dimensions as shown in Figure 4. From the plot using PCA to reduce dimensions, although there are a lot of overlapping among the points, but we may observe that the HEALTHCARE GROUP is in the middle upper position(hell green), CONSUMER SERVICES GROUP lies in the button position(yellow). INDUSTRIAL GOODS MATERIALS GROUP is in the middle upper location(dark blue) while CONSUMER GOODS GROUP is in the middle bottom(red). This plot is much more clear than the two-dimensional representation for only TF-IDF scores(in Figure 5).

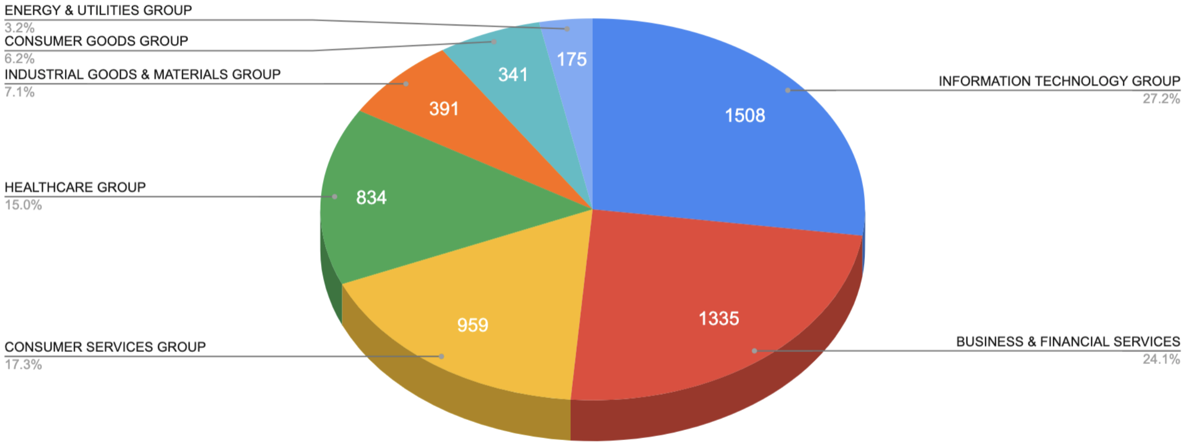


Figure 3: Websites in each category and their percentage

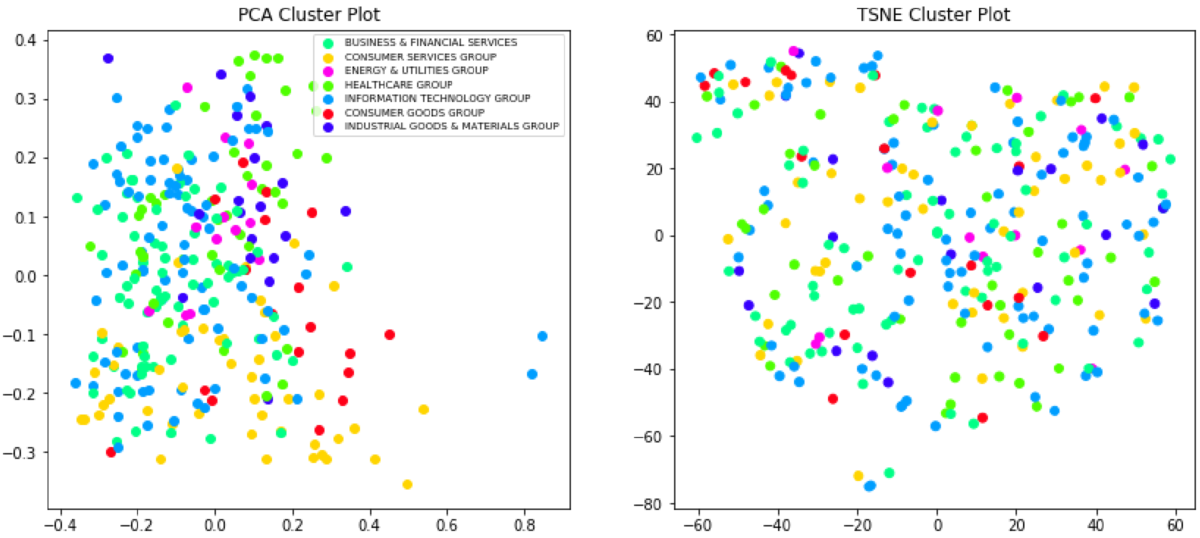


Figure 4: Plot the vectors generated by TF-IDF with Word2Vec into two dimensions using PCA and T-SNE

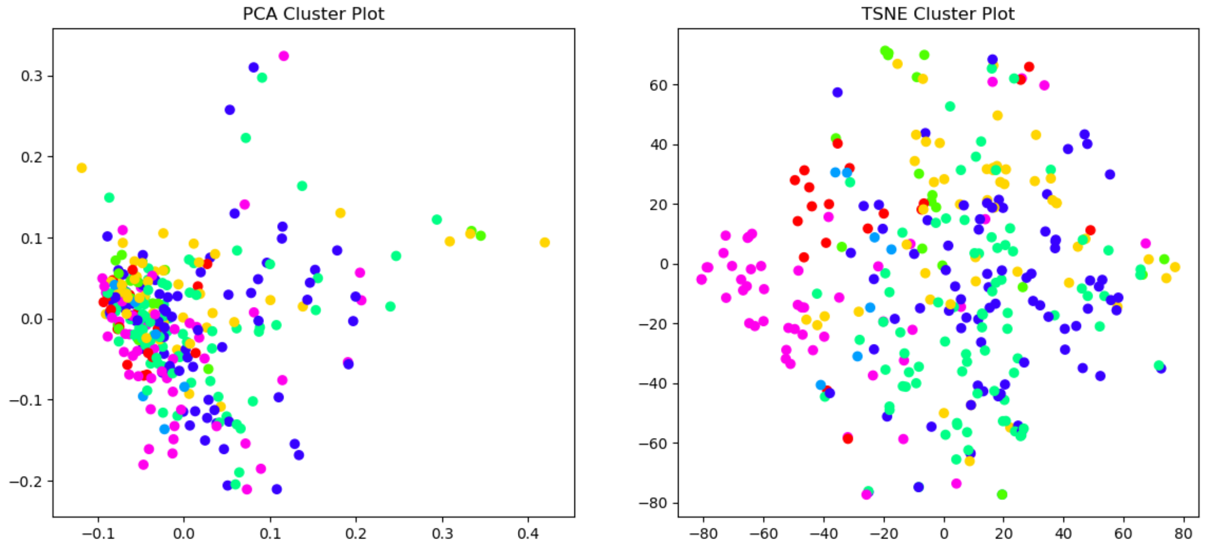


Figure 5: Plot the vectors generated by TF-IDF scores into two dimensions using PCA and T-SNE

2.2 Doc2Vec

Doc2vec is a NLP technique to represent a complete document as a vector and is the generalized version of word2vec method. In our project, we will be using the doc2vec techniques in Gensim. The process flow is as follows:

Text preprocessing →Build the Doc2Vec Model →Infer the feature vector →Build the classifier .

The *text preprocessing* step involves removing punctuation, tokenization, creating tagged content, etc. For the *Doc2Vec* step, there are two main algorithms which are Distributed Bag of Words(DBOW) and Distributed Memory(DM). In our project, we have decided to pair these two algorithms as empirical result has shown improvement in accuracy by combining these two algorithms. In the *feature vector* step, a inference training process is conducted to find a good vector to predict the website's word. Then at the last step, a SVM classifier is built to utilize the vector obtained from previous step to classify the website business categories.

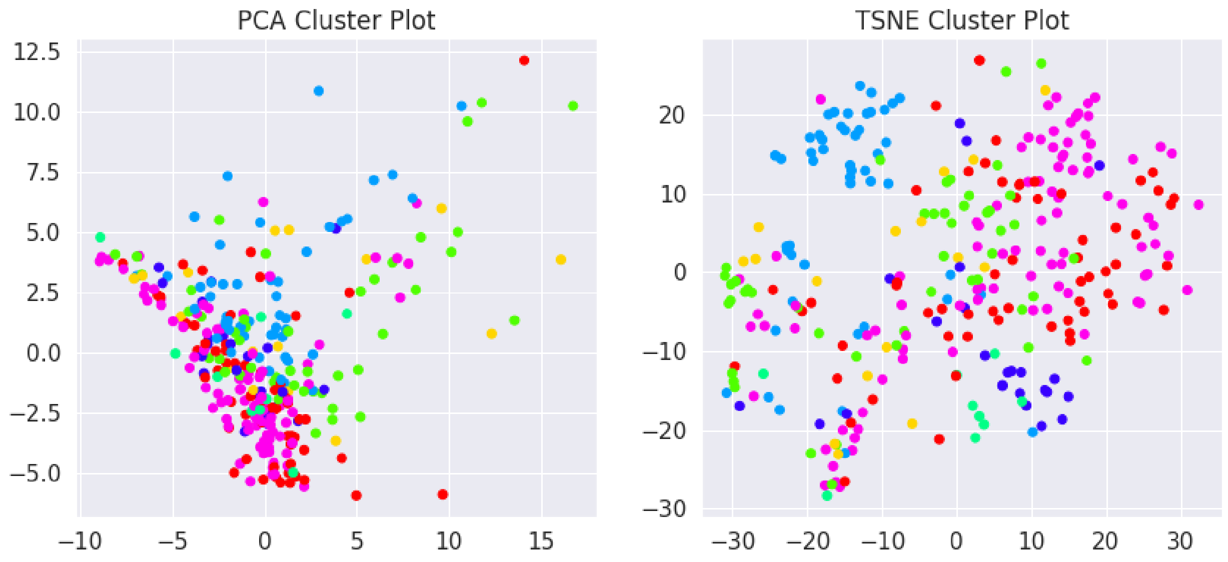
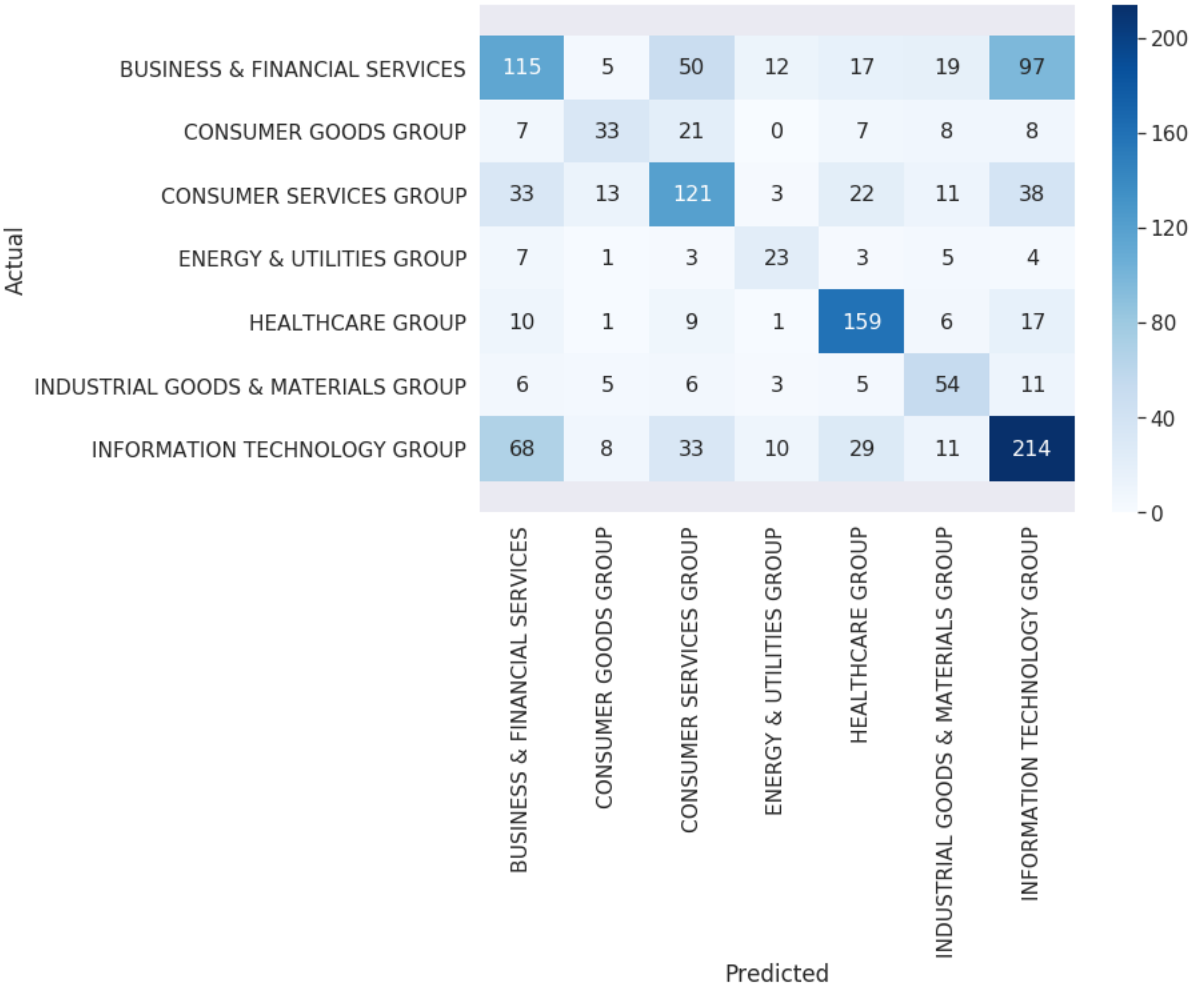


Figure 6: Plot the vectors generated by Doc2Vec into two dimensions using PCA and T-SNE

Figure 6 shows the plot of the vectors generated by Doc2Vec. It can be seen that the clustering is even clearer with this approach compared with TF-IDF and Word2Vec in the last section. On the other hand, the company categories classification result on Group 1 data is shown in Figure 7. An accuracy of 53% is achieved which is slightly lower than the TF-IDF and Word2Vec approach.



Actual \ Predicted	BUSINESS & FINANCIAL SERVICES	CONSUMER GOODS GROUP	CONSUMER SERVICES GROUP	ENERGY & UTILITIES GROUP	HEALTHCARE GROUP	INDUSTRIAL GOODS & MATERIALS GROUP	INFORMATION TECHNOLOGY GROUP
BUSINESS & FINANCIAL SERVICES	115	5	50	12	17	19	97
CONSUMER GOODS GROUP	7	33	21	0	7	8	8
CONSUMER SERVICES GROUP	33	13	121	3	22	11	38
ENERGY & UTILITIES GROUP	7	1	3	23	3	5	4
HEALTHCARE GROUP	10	1	9	1	159	6	17
INDUSTRIAL GOODS & MATERIALS GROUP	6	5	6	3	5	54	11
INFORMATION TECHNOLOGY GROUP	68	8	33	10	29	11	214

Figure 7: Confusion matrix of the classification accuracy using Doc2Vec model

3 Similarity Comparison

Text Similarity is one of the essential NLP techniques to find the closeness of different groups of texts. This allows the user to extract semantically similar questions, searching similar documents or recommending similar articles, etc. In our project, we have decide to adopt this technique to compare the similarity within different business categories as well as across them. Hopefully, this can help generate insights to find companies with similar business models. In order to perform text similarity calculation, a classic process flow is:

$$\text{Text preprocessing} \rightarrow \text{Feature extraction} \rightarrow \text{Vector similarity calculation} \quad .$$

As mentioned in previous reports, *text preprocessing* step includes removing html tags, removing stop words, etc. *Feature extraction* can be done in several ways such as TF-IDF, word2vec, doc2vec, etc while *vector similarity calculation* can be achieved by using euclidean distance, word mover's distance, cosine similarity, etc. Among them, our project decided to use the cosine similarity method, which is also the most widely used vector similarity calculation method.

3.1 Similarity with TF-IDF

Similarity matrix for the feature vectors extracted by TF-IDF is shown in Figure 8. It can be observed that the diagonal values are the biggest for each row and each column. This indicates that the similarity within the business categories is always higher than across the categories. However, it is also noticed that the similarity values are relatively low with a feature vector of size 8000 for each website. This is because that the similarity scores decreases as the feature vector size increases and the magnitude of the tfidf value of each word doesn't impact much the cosine similarity.

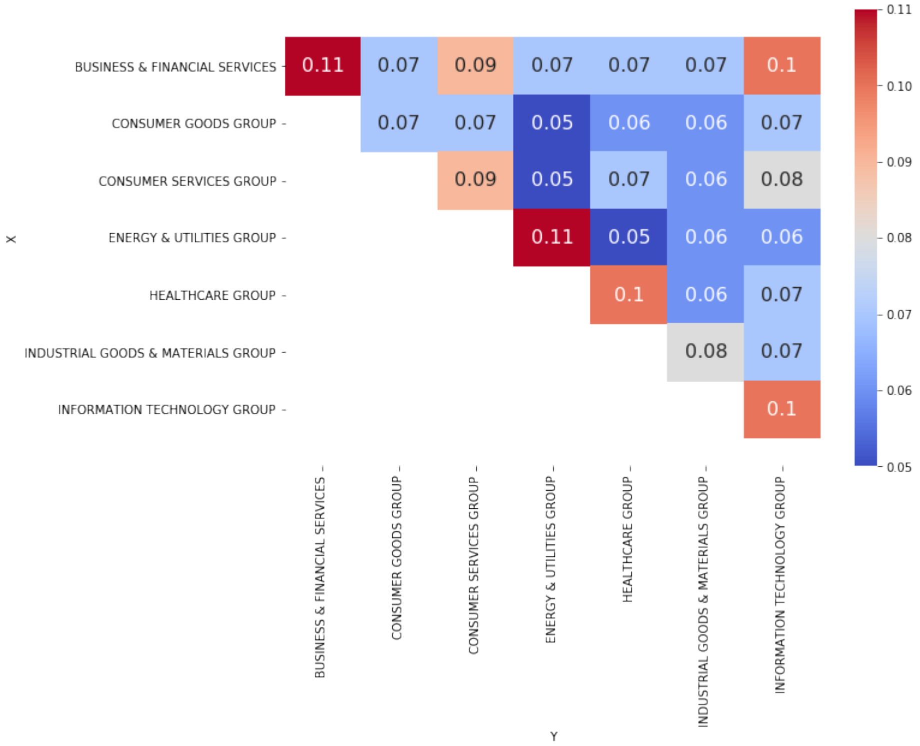


Figure 8: Similarity Matrix of TF-IDF

3.2 Similarity with Word2Vec

Top 100 tf-idf words were selected before vectorization because top 100 tf-idf gave the best accuracy as mentioned above. The words were vectorized by GoogleNews-vectors-negative300[goo, 2013]. The average word2Vec values of all words were calculated in each website. We sorted all categories and compared them by computing their similarity.

In the similarity matrix, the diagonal entries are the optimal value in each row or column. So it can concluded that each category always has the most similarity compared to itself.
Once a new website appears, we can compute its similarity with each category respectively. Except for BUSINESS FINANCIAL SERVICES and CONSUMER SERVICE, the category of the website can be estimated by checking the largest similarity of the category it compared with.

3.3 Similarity with Doc2Vec

Similarity matrix for the feature vectors extracted by Doc2Vec is shown in Figure 10. It can be observed that the diagonal values are again the biggest for each row and each column. Moreover, it seems also to suggest that BUSINESS FINANCIAL SERVICES and INFORMATION TECHNOLOGY GROUP has a close relationship as the similarity values across these two categories are very close to the similarity values within their categories.

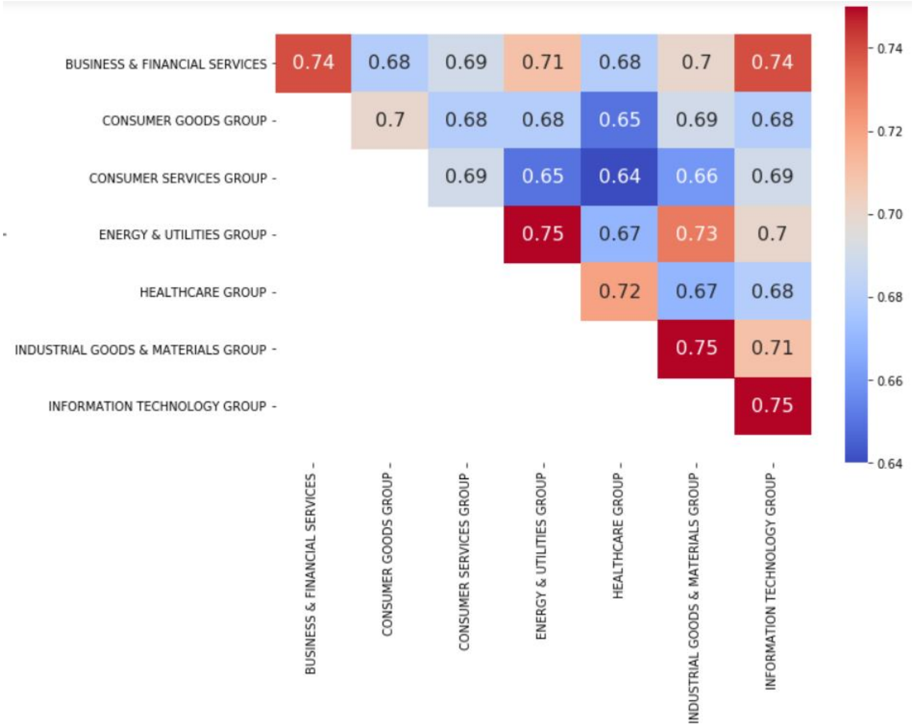


Figure 9: Similarity Matrix of word2vec

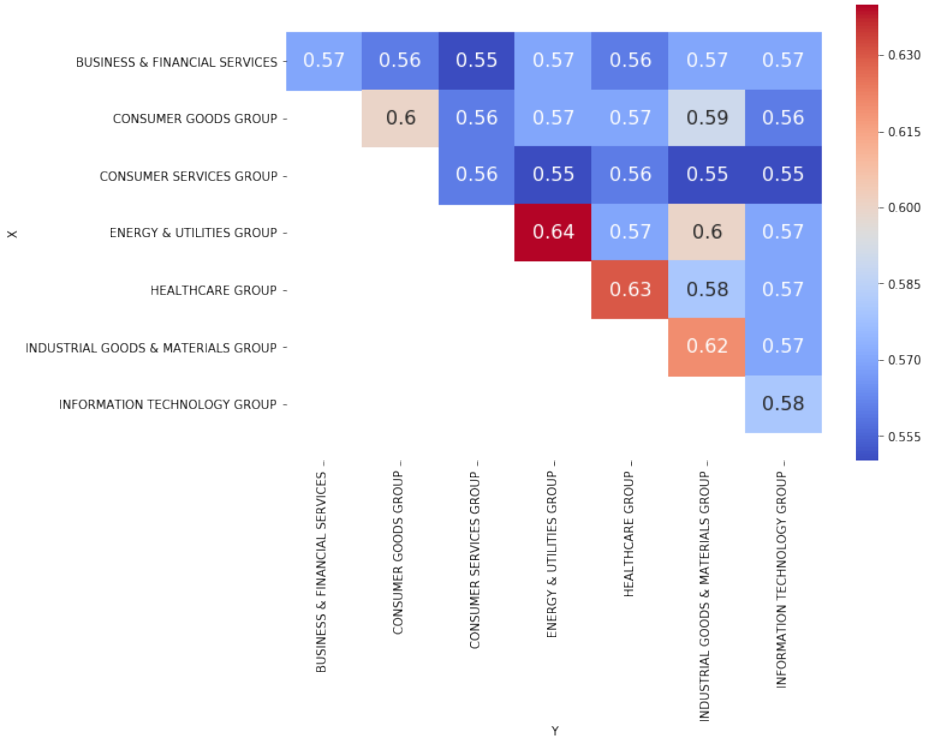


Figure 10: Similarity Matrix of Doc2Vec

4 Next Steps

For the next steps, we will focus on analysis the websites inside one category in January using unsupervised method and refactor the codes as well as finish the documents in February.

References

Google code archive - long-term storage for google code project hosting., Jul 2013. URL https://code.google.com/archive/p/word2vec/.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.