The University of Hong Kong
HKU Business School
IIMT4601 Information System Project Management
Semester 2, 2020-2021

Project Report:

Enhancing Clustering for Société Générale's News Digest System

Date of Submission: 7th May 2021

Table of Contents

1. Executive Summary

1. What is this project about?

This project is about conducting research on various NLP algorithms, in particular sentence embedding and Clustering, and hence producing a proof-of-concept for Société Générale's (hereinafter referred as 'SG') news digest system.

2.Who is our project client?

We are pleased to have the Risk Management (Market Risk) of SG (Asia) as our project client. During the project life, the group has been well-supported and guided by Mr. Ray WONG (Digital Officer), Miss Yiming FU (Data Scientist) and others from SG.

3. Who is our supervisor at HKU?

As this project is part of the graduate requirements of the Major in Information Systems at HKU, we have been guided and supported by Dr. Michael CHAU, associate professor at HKU Business School and the course convenor of Information Systems Project Management.

4. What is the project workflow in short?

After delineating the scope of the project in the project proposal, approvals were obtained from both SG and HKU. The group started its research on various NLP algorithms and evaluation metrics. Through several stages of testing, evaluation, tweaking and consultation with clients, our pipeline, which connects to the database of news datasets, is demonstrated and visualized on a website.

5. How to access the project deliverables?

One of the major deliverables is our website. You may access it at http://hkusg.herokuapp.com/.

6. How did the group present its findings?

The project group has delivered a total of 3 presentations. First, a prototype presentation to HKU in late March; Second, a substantive presentation to SG on 22 April where we were honoured to have the deputy head of Risk Management Asia and others joining us; Third, a final presentation to HKU on 26 April.

2. Project Outline

2.1 Background

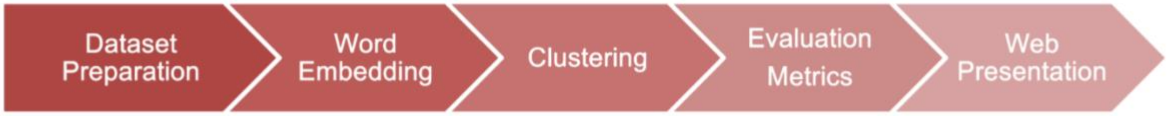
Every day in SG, analysts would read news from the daily market digest, consisting of information such as financial news and regulatory affairs across different regions. The daily market digest can allow readers to quickly grasp the latest market information and is widely used by top management executives. However, it normally takes 3-4 hours on average to produce those reports. The Design and Digital Office (DDO) of SG would like to see how to make this process more efficient. This project can assist the team in grouping news with similar headlines or topics before that stage, so as to streamline the workflow and save analysts' time.

Throughout the past few months, we have been meeting our project client bi-weekly for updates and evaluation. After hearing their insights, we identified three pitfalls in the current system.

1. Not able to group similar news headings into a group
2. Cannot identify the most popular news over a specified period
3. A lack of numerical and systematic way to evaluate the grouping results

2.2 Project Scope

After understanding the existing pitfalls, our project scope is confined to these five aspects:



2.2 Project Deliverables

Deliverables	Description
Deliverable 1: Collection of custom and non-custom datasets	Non-custom datasets refer to datasets openly available for NLP research. Custom dataset refers to the set of news articles extracted based on designated keywords
Deliverable 2: Research and Development on Word Embedding Techniques and Clustering Algorithms	
2.1 Research on various Word Embedding Techniques and Clustering Algorithms	Understanding the principles behind popular word embedding techniques that are widely used by NLP applications
2.2 Evaluation of the chosen techniques	Devising or leveraging a scientific approach to evaluate the effectiveness of different techniques and choose the technique best fit for SG

2.3 Implementation	Adopting an existing library to implement the chosen technique, or to train the model from scratch if there is no existing library
Deliverable 3: Clustering Result Evaluations	
3.1 Exploration of different evaluation metrics	Studying extrinsic and intrinsic evaluation metrics and making adjustment according to the use case
3.2 Result evaluation fixed number of clusters	Comparing the results under an equal number of clusters with custom dataset and 20NewsGroup
3.3 Result evaluation with generalized threshold	Setting a threshold value t to retrieve all the clusters at distance value t with custom dataset
3.4 Demonstration of Named Entity and Noun Extraction	Extracting keywords, including named entities and nouns, with pretrained model Spacy on top of Clustering
3.5 Result Evaluation with recursive Clustering	Applying the three approaches of recursive Clustering to yield a better result
Deliverable 4: Presentation of the findings	
4.1 Website deployment for interactive evaluation	Deploying a website connected with the implementation from deliverable 3, displaying the results of the search query and allowing users to evaluate the results interactively with network graph.
4.2 Written documentation	Producing documentation of the findings, which includes the technical details of the chosen algorithms, methodologies and results of evaluations, as well as their limitations.

3. Current Market Offerings

For research purposes, we have analysed two leading market players that best leverage Natural Language Processing techniques on news data.

3.1 Google News

Description: Google News is a news aggregator service developed by Google. The "Full Coverage" feature in Google News tried to group stories from different sources.

Rationale of adopting as reference: Our group would like to understand more about how Google News has its articles text embedded without any unwanted advertisements, copyright statements or boilerplate (i.e., noise). Most importantly, we want to learn how it selects words that are most important to the articles. For example, using "named entities" to characterize an incident or event and give them more weightage. We believe by studying the case of Google News, it can help improve the quality of our word embedding results.

Moreover, we seek pretrained word embeddings that could provide a dataset covering a much larger volume of rare words. So that the possibility of arriving at the right representation of the unannotated text would increase. To do so, it would require a rich vocabulary base. Google's news word embedding methods are trained on large text corpuses with 100 billion words (Rezaeinia et al., 2017). Its latest published word embedding method BERT is the latest evolution. It reads the text in both a traditional left-to-right manner and an unconventional right-to-left manner to better understand the meaning. Although such bidirectional training converges slower than the traditional method, it outperforms its alternatives after a small number of pre-training steps.

3.2 Bloomberg

Description: Bloomberg is the leading player in providing financial news and data. Their major product, Bloomberg Terminal, has incorporated a feature to cluster and summarize the results upon ad-hoc user queries (user search).

Rationale of adopting as reference: Bloomberg's NSTM clusters related news stories together, ranks the most important themes and subsequently provides a succinct summary for each cluster. There are two clustering stages. Stage one involves online incremental clustering at story ingestion time, which can reduce the computational workload in later stages at a lower cost. Stage two involves Hierarchical Agglomerative Clustering (HAC) at query time. Besides, it adopts recursive clustering to reduce further fragmentation, in case there are similar clusters left un-clustered.

Overall, its use-case is the most similar to what we would like to achieve: to provide clustered news headlines for users to read. Therefore, their clustering methods and procedures are good referencees for us to look into. We will include some similar features such as the provision of cluster size, links to the original news source and time selection. On top of those, we introduce new functions of "Top Keywords" and "Top Appeared Entity", which would be explained further in later parts.

4. Datasets

Collecting sophisticated datasets is a crucial stage in any kind of NLP research. The overall goal of our dataset collection is to mimic the actual use case of SG's analysts so that the evaluation results generated with our NLP model could be more fit for the purpose. Therefore, we primarily set our target at collecting text datasets of news, preferably financial news, which is within the scope of daily digest read by analysts.

4.1 Classifications of Datasets

Our group has prepared several sets of text datasets, which can be classified into custom and non-custom datasets. Custom dataset is prepared by our group specifically to mimic the actual use case, which includes financial news and news that would come to the analysts' concerns. Non-custom datasets are datasets that are popular for NLP research and have been widely applied for the the purpose of evaluating different machine learning models.

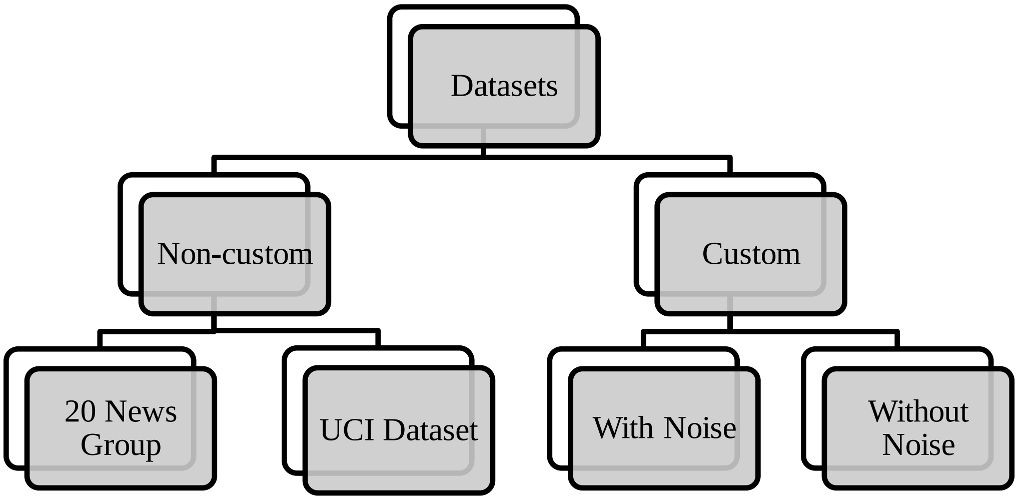


Fig. 4.1 Classifications of datasets adopted for the research

4.2 Non-custom Dataset

In the field of NLP research, there is no lack of existing datasets prepared for direct application for machine learning models, such as word embedding and Clustering. After careful examination of all the popular datasets, we have chosen 20 Newsgroup and UCI for the non-custom dataset. Both datasets concern with different categories of news articles, such as business, science, and technology. Specifically, 20 Newsgroup covers a wider range of topics including religion, sports, space, etc. Since the analysts' interests mainly lie in business-related news, the UCI dataset is more proximate to the SG use case.

The size of the dataset is another criterion of concern – the amount of data required depends on the complexity of our proposed solution and chosen algorithms. It is important to note that the project's focus is on evaluating different word embedding methods with various metrics. We are conscious that the algorithms proposed, such as BERT and USE, are existing word embedding methods. Hence, a dataset of reasonable size, say a few thousand, would suffice to correlate the performance of each algorithm with the results of different evaluation metrics (which will be further discussed in later parts of this report). After careful examination, we have adopted around 4000 articles from the UCI dataset and 1000 articles from 20 Newsgroups.

The attributes of data include the subject (news headline), date, author, and URL. Note that one of our project's deliverables is clustering articles based on its headlines. Therefore, content of the news would not be our target.

Dataset	Size	Category	Attribute

20 Newsgroup	~1000 news is chosen from 20,000 news stories	20 news groups, e.g.:
		• comp.graphics
		• rec.sport.baseball
		• sci.space
		• talk.politics.guns
		• soc.religion.christian
			Subject;
			Date;
			Text;
			etc

UCI Dataset	~4000 news is chosen from 420,000 news stories	4 categories:
		• Business
		• Science and Technology
		• Entertainment
		• health
			Headlines;
			URL;
			Category;
			Publisher;
			etc

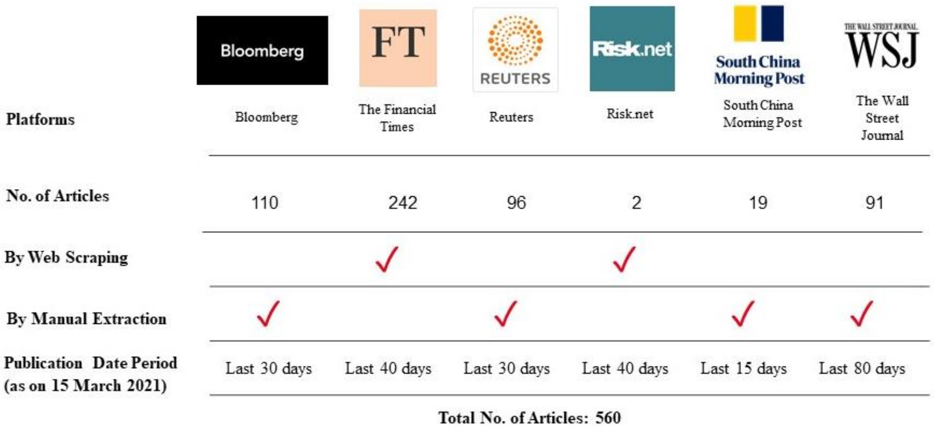
Fig. 4.2 Summary of non-custom datasets

4.3 Custom Dataset

Although the non-custom datasets are highly reliable and are of reasonable data size, we identified certain pitfalls – *first*, the sources of articles are not ones that SG analysts usually rely on. As risk analysts, they are more inclined to read news from credible sources, such as Bloomberg and The Financial Times. *Second*, as the news in non-custom datasets is not specifically provided for businessperson, the articles are not ones that SG analysts are interested in, such as popular culture. *Third*, perhaps the most compelling pitfall, we need a fixed number of pre-defined clusters for evaluation purposes. It is so required because we could draw comparison, from the clustering results, to see whether a particular word embedding method could distinguish between those pre-defined clusters (to be discussed in "Results Evaluation"). The non-custom datasets do not provide us the clear distinctions between clusters, but only the general category such as 'technology' or 'business'. It is why the project group has prepared a custom dataset for evaluation purposes.

We prepared the custom dataset by first picking some popular keywords – we have chosen a total of 5 tags, namely 'Brexit', 'Cryptocurrency', 'Electric Vehicle', 'Hong Kong', 'US-China Relations'. The reasons for selecting these tags are that, firstly, news stories relating to these tags will probably impact the financial market in which SG analysts are interested to read. Secondly, they are all recent trending news such that the collection of news would be much easier. Thirdly, news relating to these tags may demonstrate some casual relationships, for example, when Tesla (manufacturer of electric vehicles) purchases Bitcoin (cryptocurrency). Furthermore, 'Hong Kong' and 'China' demonstrate similar semantic meaning. Therefore, it is interesting to see whether each of the word embedding methods can distinguish between confusing clusters based on nouns and named entities (to be discussed).

After deciding the scope of the custom dataset, we began building it by either web scraping with Python, or by manual extraction. By either method, we retrieved the data by searching the keywords on different search engines, such as Bloomberg's, Google News's, etc. As far as the sources of news are concerned, we chose several credible news platforms, such as Reuters, Risk.net, The Financial Times, etc. Overall speaking, we were able to obtain around 560 articles across 6 news platforms. The news is all published recently, ranging from the last 15 – 80 days (as on 15 March 2021).



Platforms	Bloomberg	The Financial Times	Reuters	Risk.net	South China Morning Post	The Wall Street Journal
No. of Articles	110	242	96	2	19	91
By Web Scraping		✓		✓		
By Manual Extraction	✓		✓		✓	✓
Publication Date Period (as on 15 March 2021)	Last 30 days	Last 40 days	Last 30 days	Last 40 days	Last 15 days	Last 80 days

Total No. of Articles: 560

Fig.4.3 Summary of the custom dataset

4.3.1 Features of Custom Dataset

To enhance the reliability and difficulty of the custom dataset, in collecting the news, we incorporated the following features into the dataset. Firstly, for around one-third of articles' headlines, the wordings of the tags themsevles do not appear in the titles – for example, the title only includes Biden, but have no mention of words like US-china relations. Similarly, 'Carries Lam' instead of 'Hong Kong' and 'Northern Ireland Agreement' instead of 'Brexit' are represented in the news titles. Secondly, the use of synonyms or alternative words across various also gives to the difficulty of the dataset. For example, 'Beijing' and 'China' are inter-changeable for the purpose of news titles because both terms point to the meaning of 'Chinese government'. The same logic goes to 'Cryptocurrency' versus 'Digital Currency'. Thirdly, the use of abbreviations, such as 'EV', 'HK', 'US', etc., also adds difficulty to the dataset. Lastly, we have included some barely related news. For example, some news titles have no mention of 'Brexit' at all. But nonetheless, they include terms that suggest a minimal relationship with Brexit, such as Amsterdam, and Scottish Independent Movement – they only barely related to Brexit. Overall, we have ensured that the custom dataset is of reasonable difficulty for our evaluation purposes.

4.3.2 Noisy Data

As suggested by our project client, we have added some noisy data in the same set of custom dataset in furtherance of the clustering result evaluation. Noisy data is data that is meaningless, or unstructured or cannot be understood and interpreted correctly by machines. Based on this criterion, we have incorporated a few types of noisy data.

First, we introduced spelling errors to some of the news articles. We replaced some terms with words of entirely different meaning, such as replacing 'Finance' by 'Fiancée'. Such exercise will allow us to compare the evaluation results and to see if the pre-trained models will still be able to pick up the mistaken words. Second, we introduced some uncommon abbreviations, such as 'ECJ' (European Court of Justice). Third, we introduced articles that contain slang, such as 'Wolf Warrior', a term to describe the aggressive Chinese style of diplomacy.

5. Technical Specifications

In this section, we are going to go through the different steps in developing the clustering algorithm for the project.

5.1 Clustering
5.2 Word Embedding
5.3 Named Entity Recognition
5.4 Enhancement
5.5 Conclusion

5.1 Clustering

While there are several available clustering methods, we select Hierarchical Agglomerative Clustering (HAC) for our pipelines upon comparison.

This report will only explain some of the considered yet abandoned methods under our considerations, namely DBSCAN and OPTICS.

In our initial selection, we only shortlisted some of the clustering methods for second-round comparison based on their general performance in distinguishing clusters.

Below is a graph showing a general comparison of a variety of methods in assigning objects in different configurations to their respective clusters.

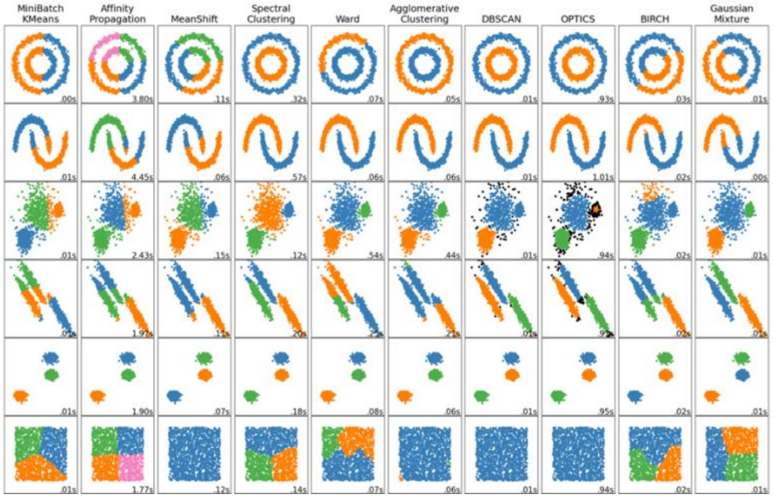


Fig.5.1 Comparison of Performance of various Clustering Methods (Scikit-Learn, n.d.)

From the graph above, it is noticed that Hierarchical Agglomerative Clustering (HAC), DBSCAN and OPTICS can consistently identify objects of different configurations into well-separated clusters.

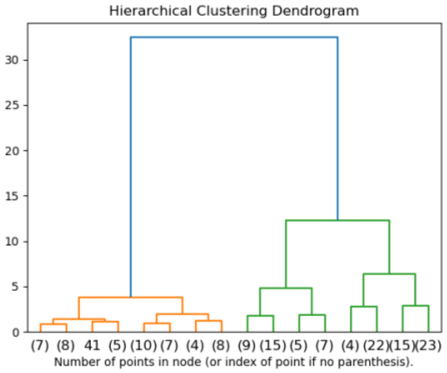
For the dataset in the first row, only the aforementioned three methods and Spectral Clustering successfully identify the objects into inner and outer circular clusters. Yet, for the law row's dataset, Spectral Clustering fails to assign all data points to a single cluster. As a result, the above three methods are shortlisted as the final candidates.

5.1.1 Summary of final candidates

The final three candidate methods are briefly introduced.

5.1.1.1 Hierarchical Agglomerative Clustering (HAC)

This algorithm adopts a bottom-up approach: each initial observation (raw data). Start in its own cluster at the root, and they are merged successively for several loops into a smaller number of nested clusters. One layer is built in the hierarchy of the clusters after each loop. Ultimate all nested clusters are merged to one single cluster. It is visualized into a dendrogram below with each vertical line represent one cluster, and the y-axis measures the distance of the clusters from the original observations.



5.1.1.2 DBSCAN

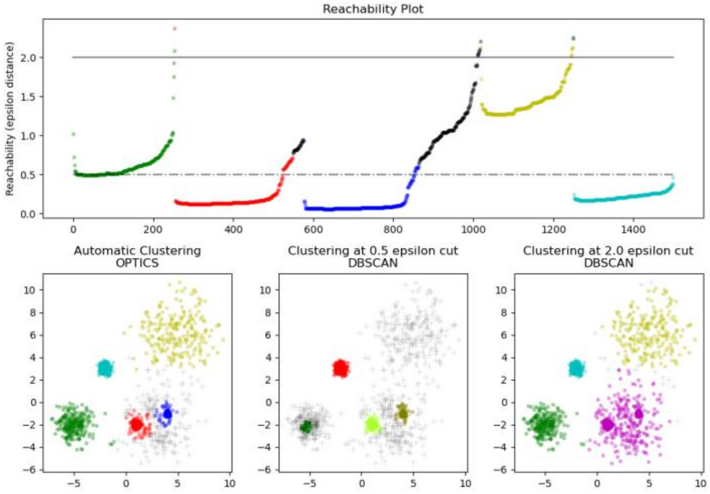
This algorithm centred on the perception of clusters as areas of high density separated by areas of low density. To be exact, the cluster is defined as a set of core samples (i.e. samples in areas of high density measured by some distance measure), and a set of non-core samples that are close to a core sample (but are not themselves core samples) as the fringes of a cluster.

It is noted that the parameter 'eps' (epsilon) needs to be *appropriately* chosen for the data set and distance function. It represents a maximum distance from core samples within which other non-core samples are defined to be neighbours of the core sample. If chosen too small, most data will be classified as noise. If chosen too large, eventually the entire dataset will be returned as a single cluster. Heuristics is needed to search for an appropriate eps.

5.1.1.3 OPTICS

This algorithm shared similarities with DBSCAN and can be regarded as a generalization of DBSCAN that relaxes the *eps* requirement from a single value to a value range, known as *reachability* distance, by setting a maximum epsilon *max_reps.*

On the right is a reachability plot where Y-axis represents point density (epsilon distance) and points are ordered such that nearby points are adjacent. 'Cutting' on the reachability plot at a single value (*max_reps*) produces DBSCAN-like results, with all points above the 'cut' are classified as noise.



5.1.2 Verdict: Hierarchical Clustering

We have set two criteria for the comparison among final candidates, namely (1) compatibility with cosine similarity metric and (2) ease of use and implementation.

5.1.2.1 Criteria 1: Cosine Similarity Metric

Before Clustering, the news headlines are converted into a vector using chosen word embedding methods (to be explained in section 5.2). In such semantic analysis, we intend to discard the effect of the magnitude of the vectors, also known as the length of the headlines, in measuring the similarity of two headlines. Therefore, cosine similarity is used as the distance metric.

Our objective is to merge similar headlines into one cluster if they are referring to the same *topic*. For example, two headlines discussing US-China relations should be grouped as one cluster. Yet, given different conventions for various news agencies, the length of headlines and frequency of word appearance varies. If Euclidean distance is used, the headline with "China" appearing twice will be more related to the topic of US-China relations. Yet, in fact it is just as equally related as the headline with "China" appearing only once. As a result, cosine similarity is used to discard the magnitude effects.

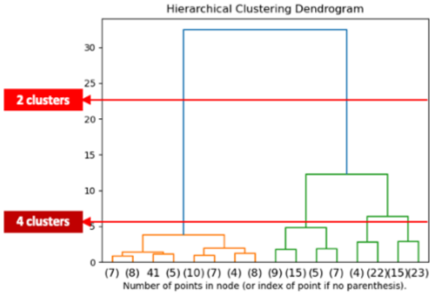
Hierarchical Clustering accommodates any pairwise distance, including cosine distance metric, while DBSCAN and OPTICS only permit distances between nearest points. In theory, the cosine metric can be used in all three methods. Yet, the application of cosine metric in DBSCAN and OPTICS could yield difficulty in implementation, which is the focus of the following part.

5.1.2.2 Criteria 2: Ease of Use

Given the time limitation of this project, the ease of implementation of the methods must be taken into account. Generally speaking, the easier the method can be deployed, the shorter time we need to integrate it into our pipelines, the higher rank in our choice of methods.

The ultimate difference between the final three candidate methods lies in the ease of determining the number of clusters. Too few or too many clusters will render the Clustering meaningless as it fails to organize the scattered information in numerous observations.

Hierarchical Clustering is the easiest to determine the number of clusters. With the aid of dendrogram visualization, simply by deciding the value of the linking threshold (i.e. the horizontal line cutting the dendrogram in the graph on the right), we can select exactly how many clusters to be extracted in the results.



On the other hand, controlling the number of clusters is more difficult in DBSCAN and OPTICS. The number of clusters to be obtained cannot be directly determined by selecting a definite value for the parameter, but only by repeatedly testing different values for the major parameters (*min_samples*, *eps* and *max_eps*) and adjusting after comparing the results. Such trial-and-error processes will be required for each new dataset and are time-consuming. In such a case, evaluating the pipeline's efficacy with multiple datasets become nearly impossible for our project, which is undesirable.

On top of ambiguous parameters, the use of the cosine metric complicates the implementation. Given that cosine distance is bounded by [0,1], the tolerance distance *(eps*) should be a tiny value and need a data type with double precision (np.float64) instead of ordinary float (np.float32) (StackOverflow, 2015). The trial-and-error of *eps* becomes even more difficult, and so as the adoption of the DBSCAN method.

5.1.3 Mini-Conclusion

	Hierarchical Clustering (Adopted)	**DBSCAN (Abandoned)**	**OPTICS (Abandoned)**
Geometry (Metric)	Any pairwise distance – including cosine distance	Distances between nearest points – not friendly with cosine distance	Distances between points
Ease of Use	**Easy**. Directly set a parameter to decide no. of clusters with the aid of graph	**Difficult**. Repeated testing of parameters to fine-tune the no. of clusters retrieved	**Difficult**. Repeated testing of parameters to fine-tune the no. of clusters retrieved

Having discussed the compatibility with cosine similarity and ease of implementation, Hierarchical Clustering is chosen as our final method because it seamlessly accommodates cosine metric, with no extra difficulty of adoption added. Also, it is relatively easy for us to deploy and control the number of clusters to be extracted. Once we find an optimal value for the linkage threshold, it can be universally applied across different datasets without case-by-case fine-tuning.

Apart from these clustering algorithms, there are other considered algorithms that is worth to mention. The notable one is *Latent Dirichlet Allocation (LDA)*. LDA is a topic model algorithm used to classify text into a particular topic. It may either identify a topic per document or map a list of words per topic. There are two reasons for not adopting this model in this project.

First, the output of LDA is a mapping of topics for *each* document instead of predicted clustering assignments. This algorithm does not group articles together. It does not stand in line with our primary objective of merging numerous news articles into a digestible number of news clusters.

Second, it needs training and disregards syntactic information, i.e., treating the sentences as bags of words. Since we are using news headlines datasets, a few words for each headline might not be sufficient to train the LDA to recognize the topics per headline, because topic modelling relies on counting words and grouping similar word patterns to infer topics within unstructured data.

5.1.4 Hierarchical Clustering Deep Dive

Hierarchical Clustering does not just triumph over other methods in our comparison, but it is indeed endorsed by the industry leader in analytics, Bloomberg. Bloomberg discloses that their latest function Key News Themes (NSTM) is also using Hierarchical Clustering along its pipelines (Bambrick, 2020). The NSTM function of Bloomberg has a lot of resemblance to our proposed deliverable and is worth assimilating for two reasons. First, Bloomberg is costly and not readily available for all analysts. Second, customization is not possible on Bloomberg Terminal. If we can subsume Bloomberg's state-of-the-art techniques into our project, our deliverable could serve some reference for SG in developing its homdemade and customized NLP pipelines. To sum up, Hierarchical Clustering is a proven method in the industry, which reinforces our decision to choose this method.

5.1.4.1 Linkage

Linkage refers to the definition of distance in Clustering. Hierarchical Clustering supports single, average, Complete and Ward linkage. Their definition is given:

- 'ward' minimizes the variance of the clusters being merged.
- 'average' uses the average of the distances of each observation of the two sets.
- 'complete' linkage uses the maximum distances between all observations of the two sets.
- 'single' uses the minimum of the distances between all observations of the two sets.

'Ward' linkage is first discarded from consideration as it supports Euclidean distance only while our project is using the cosine distance metric.

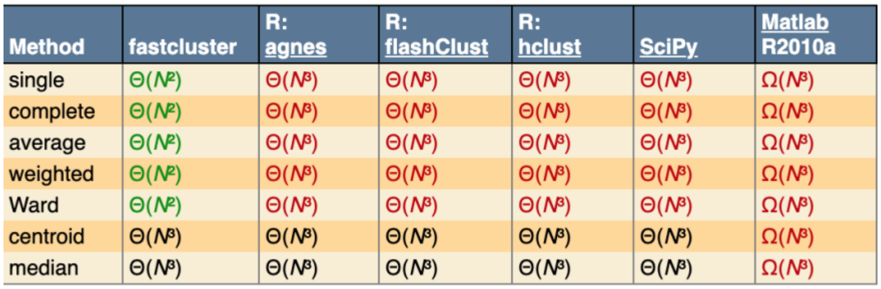
'Single' linkage is not chosen too because of its "chaining effect" where the two most dissimilar clusters can be grouped as one by considering only the *nearest* observations across two clusters. Thus, we cannot demarcate news of two distinct topics into separate clusters.

Considering the target readers of this project are the analysts in an investment bank, the Clustering should be ancillary in nature to reduce their workloads, instead of complicating their work. It is therefore a prioritized goal to reduce the number of unsuccessful Clustering (Clustering with irrelevant news grouped together). 'Complete' linkage is preferred because it can minimize the distance of the *farthest* observations within the same cluster. In other words, it minimizes the irrelevance of the two most irrelevant news within a cluster. 'Complete' linkage performs better than 'average' linkage in creating congruent clusters (Bloomberg, 2020). As a result, 'complete' linkage is chosen for this project.

5.1.4.2 Choice of Libraries

There is a number of libraries for implementing hierarchical Clustering. We choose Scikit-learn, which is a python module built on top of SciPy, as it is easy to use and multifunctional (for instance, it includes evaluation metrics). A detailed manual and instructions are available online to guide us through the implementation using Scikit-learn.

However, we are aware of the alternatives which could be faster than Scikit-learn, known as the Fastercluster. Below is the asymptotic worst-case time complexity of various libraries (Müllner, 2013).



Method	fastcluster	R: agnes	R: flashClust	R: hclust	SciPy	Matlab R2010a
single	$\Theta(N^2)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Omega(N^3)$
complete	$\Theta(N^2)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Omega(N^3)$
average	$\Theta(N^2)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Omega(N^3)$
weighted	$\Theta(N^2)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Omega(N^3)$
Ward	$\Theta(N^2)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Omega(N^3)$
centroid	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Omega(N^3)$
median	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Theta(N^3)$	$\Omega(N^3)$

Given that the Fastercluster package is a C++ library while our team's skills sets are surrounding Python, we decide to continue our implementation using Scikit-learn.

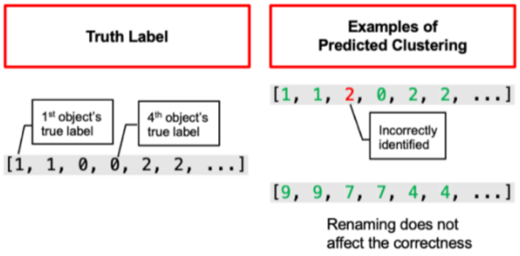
Since Bloomberg is adopting Fastcluster because of the computational speed (Bloomberg, 2020), we also recommend SG to integrate this project with Fastercluster in future development.

5.1.5 Evaluation Metrics

A basket of evaluation metrics is deployed to help us to evaluate the quality of our clustering systematically and objectively. They are assessing different aspects of clustering and offer us comprehensive benchmarks when we are comparing various word embedding methods and evaluate the robustness of our pipeline.

5.1.5.1 Extrinsic Metrics

Extrinsic metrics are characterized as the metrics that require a ground truth label, known as the correct class of the observations in the dataset. For instance, in our project, the truth label is a list of correct categories of all news headlines. As demonstrated from the graph on the right, extrinsic metrics cross-examine the truth label and predicted clustering assignments, while renaming does not affect the correctness.



5.1.5.1.1 Adjusted Rand Index

Rand Index measures the similarity between the ground truth label and predicted tags from Clustering. C is a ground truth class assignment, while K is the clustering tags. *A* and *b* are defined as:

- *a*, the number of pairs of elements that are in the same set in C and in the same set in K
- *b*, the number of pairs of elements that are in different sets in C and in different sets in K.

The unadjusted Rand Index (RI) is given by:

$$RI = \frac{a+b}{C_2^{n_{samples}}}$$

where $C_2^{n_{samples}}$ is the total number of possible pairs in the dataset. This score is bounded by [0,1]. Yet, the unadjusted index does not guarantee that random label assignments will a value close to zero.

Therefore, correction is introduced by using the adjusted Rand Index (ARI):

$$ARI = \frac{RI - E[RI]}{\max(RI) \ - E[RI]}$$

which has a new bounded range of [-1,1] and random assignment should get a value close to zero. The higher the scores, the higher similarities between the truth labels and clustering assignments.

5.1.5.1.2 Homogeneity, Completeness and V-measure

These three metrics are formulated using Shannon's entropy. Homogeneity score measures whether *each cluster* contains members of a *single class* in the truth label, while Completeness score measure whether *all members of a given class* in the truth label are assigned to the *same cluster*, given their formula:

$$homogeneity = 1 - \frac{H(C\ |K)}{H(C)}\ ; completness = 1 - \frac{H(K\ |C)}{H(K)}$$

where H(C |K) is the conditional entropy of the truth classes given the cluster assignments; H(C) is the entropy of the truth classes; H(K |C) is the conditional entropy of the clusters given the truth class; H(K) is the entropy of the clusters. They can be interpreted as a measure of information loss after Clustering.

V-measure is the harmonic mean of homogeneity and completeness, given by the formula:

$$Vmeasure = \frac{(1+\beta) * homogeneity * completeness}{\beta * homogeneity + completeness}$$

where β can be varied to alter the weights attached to homogeneity score and completeness score.

They all have a bounded range of [0,1]. The higher the score, the less information loss during the Clustering and the more accurate the Clustering.

5.1.5.1.3 Contingency Matrix

A contingency matrix is used to map out the relationship between the clustering tags and ground truth class assignments. This metric is used to evaluate the distribution of news headlines of a given category over a number of clusters. It will be acceptable if some related categories (e.g. US-China Relations and Hong Kong) might be mixed together in a single cluster, while another cluster is poor if it includes news headlines from all categories (likely that all unclassified observations will be grouped in a residual cluster). Examples will be given in the result evaluation of word embedding methods in section 5.2.4.

5.1.5.1.4 Loss

On top of the Contingency Matrix, Loss is a metric that counts the number of observations assigned to poor clusters. The definition of a poor cluster is defined as a cluster whose observations lack a dominant truth class(es). For instance, the mentioned residual clusters that encompass news from all categories each of which has not taken a majority is a poor cluster. All news in that poor cluster will be counted into the 'loss' metric. Examples will be given in the result evaluation of word embedding methods in section 5.2.4.

5.1.5.2 Intrinsic Metrics

Unlike extrinsic metrics, intrinsic metrics do not require knowledge of ground truth labels for evaluation. The evaluation can be performed using the model itself.

5.1.5.2.1 Generic Silhouette Coefficient

The generic Silhouette Coefficient measure how well-defined the clusters are. It consists of the measure of cohesion within clusters and separation between clusters, given by the formula:

$$Silhouette\ score = \frac{b - a}{\max{(a, b)}}$$

where *a* and *b* are defined as:

- *a*, the average *distance* between a sample and all other points in the same cluster
- *b* is the average *distance* between a sample and all other points in the next *nearest* cluster.

It is transparent that *a* refers to the coefficient of cohesion while *b* refers to separation.

The distance accommodates any pairwise distance, including Euclidean distance and cosine distance. It has a bounded range of [-1, 1]. The higher the scores, the more well-defined the clusters are.

5.1.5.2.2 Adjusted Silhouette Coefficient

The original author of Silhouette Coefficient proposed that modification must be made for data consist of similarities, instead of dissimilarities (i.e., distance) (Rousseeuw, 1987). In previous parts, we have explained the application of cosine similarity in semantic Clustering. While generic silhouette coefficient can still accommodate cosine distance $(= 1 - cosine\ similarity)$, it is not directly measuring cohesion and separation in terms of cosine similarity. Therefore, we modify the original formula of the Silhouette score according to the original author's proposal and adopt cosine similarity as a metric:

$$Adjusted\ Silhouette\ score = \frac{a - b}{\max{(a, b)}}$$

where *a* and *b* are defined as (modification underlined):

- *a*, the average *<u>similarity</u>* between a sample and all other points in the same cluster
- *b*, the average *<u>similarity</u>* between a sample and all other points in the *<u>farthest</u>* cluster.

The adjusted Silhouette Coefficient shall offer a more tailor-made evaluation for our Clustering using cosine similarity. Same as the generic score, it has a bounded range of [-1, 1] and the higher the score, the better defined the Clustering.

5.1.6 Summary

The eight metrics to be deployed are listed below.

	Adjusted Rand Index	Homogeneity Score	Completeness Score	V Measure	Contingency Matrix	Loss	Generic Silhouette Coefficient	Adjusted Silhouette Coefficient
Truth Label needed?	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Output	[-1, 1]	[0, 1]	[0, 1]	[0, 1]	-	-	[-1, 1]	[-1, 1]

5.2 Sentence Embedding

After deciding the clustering algorithm, sentence embedding component is another important piece for an NLP task. Sentence embedding is the technique to map sentences from sentences to vectors of real numbers, it is required for our task because any clustering algorithm would require the input to be a numerical value.

In this section, we will discuss,

1) The different sentence embedding models we have considered
2) Choosing the models for the project
3) Implementation of the models
4) Results evaluation

5.2.1 The different sentence embedding models we have considered

By the definition of sentence embedding, essentially there are infinite methods for us to map sentences to numerical values. For example, a one hot encoder serves this purpose as well. While these methods were effective in simple text-processing tasks, they did not really work on the more complex ones. It was by 2013, the discovery of Word2Vec (a word embedding model), that changed this landscape, Word2Vec successfully converts the word into a vector that able to identify the semantics and syntaxes of the word. This path-breaking innovation in word embedding has quickly scaled to many different tasks including sentence embedding. We evaluated 6 sentence embedding models, from the most traditional to the most advanced technique, namely TF-IDF, Average of Word2Vec, Doc2Vec, Autoencoder (NVDM), USE, BERT.

5.2.1.1 Terms Frequency – Inverse Document Frequency (TF-IDF)

Before understanding TF-IDF, we shall give an introduction of the most commonly used methods, Bag-of-words (BoW). It built a dictionary that contains every word appearing in the documents, and each document is a representation of the occurrence of words. TF-IDF is a simple twist on the BoW, yet it looks at a normalized count where each word count is divided by the number of documents.

Definition of TF-IDF:

$$tf(w,d) = \#\ times\ word\ w\ appears\ in\ documentd$$

$$idf(w,d) = \frac{\#\ Documents}{\#Documents\ contains\ w}$$

$$tf\text{-}idf(w,d) = tf(w,d)\ ´\ idf(w,d)$$

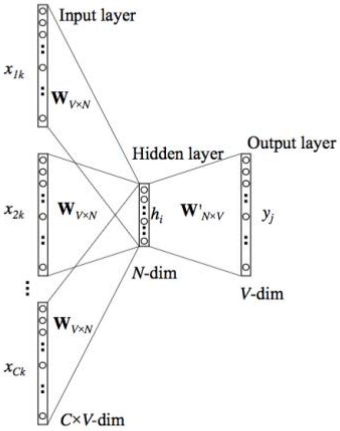
From the definition we can learn, the more frequently a word appears in multiple documents, its idf function (Inverse document frequency) is closer to 1, vice versa its idf function is much higher. Thus, TF-IDF compares to BoW, makes rare words more prominent and effectively ignores common words.

Although TF-IDF is very simple to understand and easy to implement, its shortcomings can be easily spotted. First, its dictionary is a fixed set of words of the training dataset and cannot cover the unseen words in reality, causing a loss of accuracy. Secondly, the dictionary is creating a sparse representation which is hard to model for a computational reason. Thirdly, the normalized count of occurrences lost the context and word order, these elements matter a lot in language processing tasks. Also, the model does not know how to differentiate synonyms, or n-gram pattern of words, etc. Thus, we need a more advanced solution than TF-IDF.

5.2.1.2 Average of Word2Vec

Word2Vec is a word embedding technique developed by Google in 2013. The fact that how Word2Vec leveraged neural networks to make it capable of capturing a word's context, semantic and syntactic similarity, and relation with other words has brought a significant impact to the industry. To further understand how it achieves that, let us take a look at its architecture.

Fig 5.2.1.2 Architecture of Word2Vec



Word2Vec used a shallow neural network that only consists of three layers, the input, one hidden and the output. The concept to train this network is based on the words around a target word, we input the two words before and after a target word as the input, and training the model to output the target word. This is known as Continuous Bag of Words (CBOW), which is why we can see there is C*V dimension in the input, C representing how many words around the target to be inputted (Mikolov, 2013). After training a large corpus of text, the weights connecting between the hidden layer and each output node are the word embeddings. Another method of Skipgram is similar and like a flipped version of CBOW, will not be discussed here.

By understanding the theory of Word2Vec, we can understand why it is able to capture the semantic similarity. A simple example such as

"Have a great day." "Have a good day"

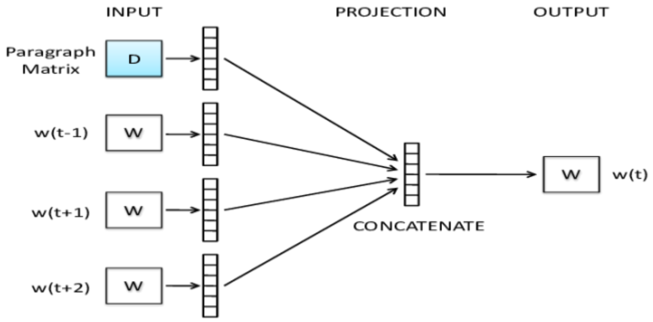
where the underlined word is the target word. Since both target words will have the exact same input, after multiple trainings with some other examples, the model will produce a high correlation between "great" and "good", thus produce semantic similarity. In fact, this kind of relation is the most unique feature of Word2Vec at that time, one could even do arithmetic operations over their output such as "King – Man + Woman = Queen".

We would like to leverage this feature from Word2Vec for our task, in order to do that we would need to find a method to combine the different word embeddings. We decided to take the average of the word embeddings as the sentence embedding. This method might not work correctly since there are many different combinations of words which taking average could lose their meaning, also this method also neglected the importance of word order, therefore it will serve as experimentation for us to try this approach.

5.2.1.3 Doc2Vec

One year after the launch of Word2Vec, an author of Word2Vec released another paper on applying the idea of Word2Vec on paragraphs/documents with a more rigorous methodology and it is then more commonly named as Doc2Vec.

Fig 5.2.1.3 Architecture of Doc2Vec



As you can see from the architecture, it is largely similar to Word2Vec except for the addition of paragraph matrix. The paragraph matrix is a vector with the length of the total number of paragraphs, each element in the matrix is relegated to a single record. During training, this paragraph matrix is also trained and the weights between each element in the matrix and the hidden layer results in the sentence embedding for that element (Mikolov, 2014).

Comparing to Word2Vec, this is specifically designed for the task of sentence embedding and for sure it has an advantage over vector averaging since its training has learnt from the entire document. However, a main limitation of Doc2Vec is its scalability. As you can spot from the above architecture, you would need to retrain the whole network every time you have a new paragraph. This would be unfavourable to our task given we do not have a rich dataset on hand and incredibly inefficient to use it in production as well.

5.2.1.4 Autoencoder (NVDM)

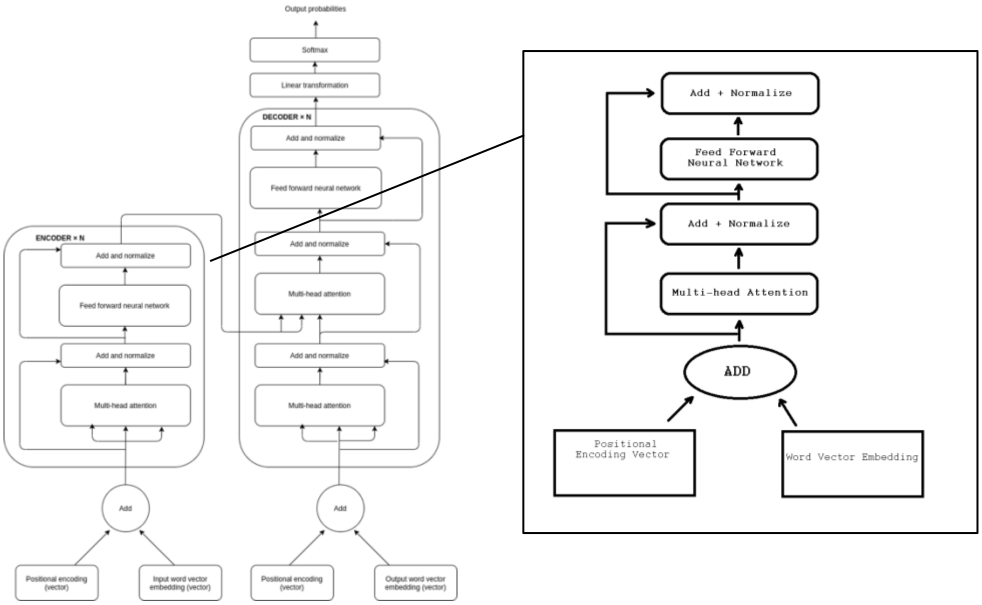
During the research of the models, we encountered the Autoencoder technique, which is a type of self-supervised model that learns a compressed representation of input data. We found it would be a clever idea to compress text into vectors, in specific leveraging LSTM to compress sequence data.

For more details of this kind of technique, we found a model named NVDM from Neural Variational Inference for Text Processing (Miao, Yu, Blunsom, 2016). This model is also employed by Bloomberg in their NSTM system, for the reason this model resembles the latent topic structure popularized by LDA which has proven effective in capturing textual semantics. (Bambrick, 2020) However, since there are not many resources about this model on the Internet and with limited expertise in this area of knowledge, we decided not to go further with this technique in this project.

5.2.1.5 Universal Sentence Encoder (USE)

Another disrupting technique is the transformer architecture that was introduced in 2017. Prior to the transformer, most language tasks rely on the Recurrent Network (RNN) type of model which uses their internal state to process variable length of sequences of inputs in order to capture the timely dependencies in sequences. Transformer, did not use any of these architectures, used the attention mechanisms that changed the whole landscape.

Fig 5.2.1.5 Architecture of Transformer



It might be intuitively complicated to look at, however if we look carefully, it is basically composed of multiple "Attention layer + Feed Forward Layer" in both Encoder side and Decoder side. These attention layers are the key changes that make a great differences. Instead of inputting a sentence in a sequence word by word, we can directly input the whole sentence and the attention mechanisms look at the whole sequence and calculate which other parts of the sequence are important for each word (Vaswani, 2017). In this way, comparing to RNN which can only capture a partial state of previous sequences, Transformer is able to capture every information in every step.

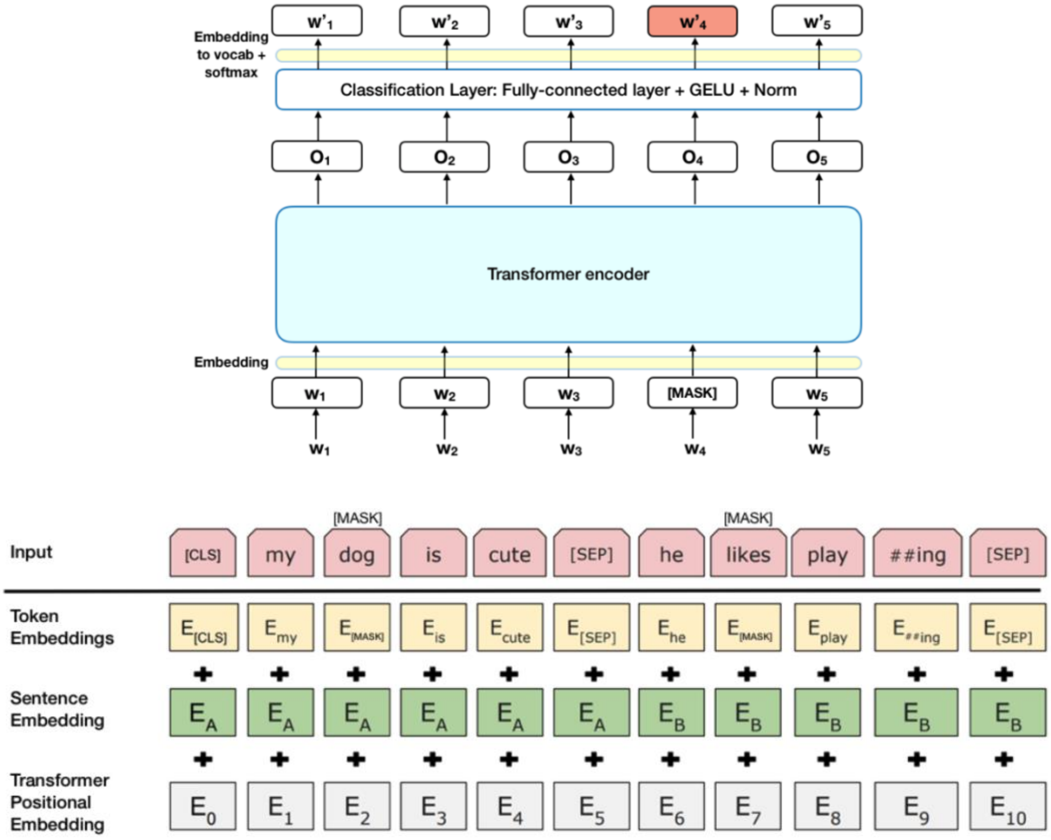
USE is one of the models that leveraged this transformer architecture and is specifically designed to do the sentence embedding task. It is trained over a number of natural language prediction tasks such as text classification, skip-thought, response prediction, etc (Cer, 2018). The goal is to train it on multiple tasks such that the same embedding has to work on multiple generic tasks, then it can capture only the most informative features and discard noise that results in a generic embedding that transfers universally to a wide variety of NLP tasks.

USE is one of the most suitable models that equipped with the latest technology and fitting the need of our use case. Unlike the use of paragraph matrix of Doc2Vec, transformer architecture allows us to input sentences directly to output the embedding which can be used for any sentences after one time of training.

5.2.1.6 BERT / SentenceBERT

BERT is the state-of-the-art model that makes use of the Transformer architecture. It comes to the leading positions among different models by designing a unique prediction goal. The training strategies of BERT consist of two parts, namely Masked LM (MLM) and Next Sentence Prediction (NSP) (Devlin, 2019).

Fig 5.2.1.6 Architecture of BERT



MLM works a little bit like Word2Vec, which masks 15% of the words in the input sequence, the model attempts to predict the value of these masked words (Devlin, 2019). NSP refers to inputting a pair of sentences, which 50% of the pairs are a subsequent sentence in the original document, and the model attempts to predict if the second sentence is a subsequent sentence or not (Devlin, 2019). That is why you can find some [SEP] token in the input from the above diagram, that indicates the separation token of sentences. Note that MLM and NSP are applied at the same time, therefore not only the [SEP] token you could find in the input, you can also find some words are labelled [MASK], which indicate it is the chosen masked word from MLM. The goal of the training for BERT will be minimizing the combined loss function of the two strategies.

Combining the Transformer architecture and this novel training strategy, BERT received state-of-the-art performance on a number of natural language understanding tasks. It is still not yet well understood how BERT achieve such outstanding performance, nonetheless it has brought a tremendous breakthrough in different language tasks. We are interested in bringing BERT to our findings to see how the latest technology in the industry could help to solve our task's problem, in specific, we employed SentenceBERT which is a variant model that built on top of BERT that specifically designed for sentence embedding tasks (Gurevych, Reimers, 2019).

5.2.2 Mini Conclusion

Having discussed the design of different models, we concluded our discussion with the following table:

	TF-IDF	Word2Vec	Doc2Vec	NVDM	USE	BERT
History	1972	2013	2014	2016	2018	2018
Author	Karen Spärck Jones	Google	Google	Yishu Miao, Lei Yu, Phil Blunsom	Google	Google
Architecture	Bag of Words	Shallow Neural Network	Shallow Neural Network	Shallow Neural Network	Transformer	Transformer
Sentence Embedding	Yes	No (Averaging Vectors)	Yes	Yes	Yes	Yes
Library Provided	Yes	Yes	No	No	Yes	Yes
Model Technology	Less Advanced	Moderate	Moderate	Moderate	More Advanced	More Advanced
Expected Performance from our team	Low	Low	Low	Medium	Medium	High
Decision to include in evaluation	Yes	Yes	Abandoned	Abandoned	Yes	Yes

For deciding whether these models will be used in our evaluation process, our team wishes to include all of them as each of them have their unique characteristics that we would love to explore how it differentiates in our task. However, given limited time and resources, we do not have the time to prepare a fair enough dataset to train these models from scratch. Therefore, whether these models are available in pretrained form from libraries become a critical factor for whether it is included in our evaluation or not. We have abandoned Doc2Vec and NVDM since there is no available pretrained model for us to test on. Doc2Vec, as we have mentioned in the introduction above, its design in nature requires new training with a new dataset which we do not have the capacity to test this model in our project.

5.2.3 Implementation

Model	Library	Remark
TF-IDF	Scikit learn	
Word2Vec	Spacy	Pretrained the same dataset from its paper
USE	Spacy	Pretrained the same dataset from its paper
BERT	Sentence Transformer	Sentence Transformer provides over 20 BERT implementations, we have picked the "paraphrase-distilroberta-base-v1" which trained on DistillRoBERTa using paraphrased data + original dataset from the paper

5.2.4 Results Evaluation

After gathering the clustering algorithm and word embedding models, we can start to build the prototype and do some evaluations. To evaluate these models, we will run them on two datasets, the 20NewsGroup and the custom dataset, and using two-parameter settings, using a fixed number of clusters and use a generalized threshold.

Dataset

20NewsGroup	Picked 50 data from each class, forming a total of 1000 data from 20 classes
Custom Dataset	560 data from the custom dataset mentioned in Section 4

Parameter settings

Fixed number of clusters	It refers to providing a maximum number of clusters to the clustering function, therefore all models will output an equal number of clusters. Comparing the results under an equal number of clusters can better understand eliminate the risk of certain threshold favouring certain models and provide a fair comparison.
Generalized threshold	It refers to setting a threshold value t to retrieve all the clusters at distance value t such that each cluster has no greater a cophenetic distance than t.

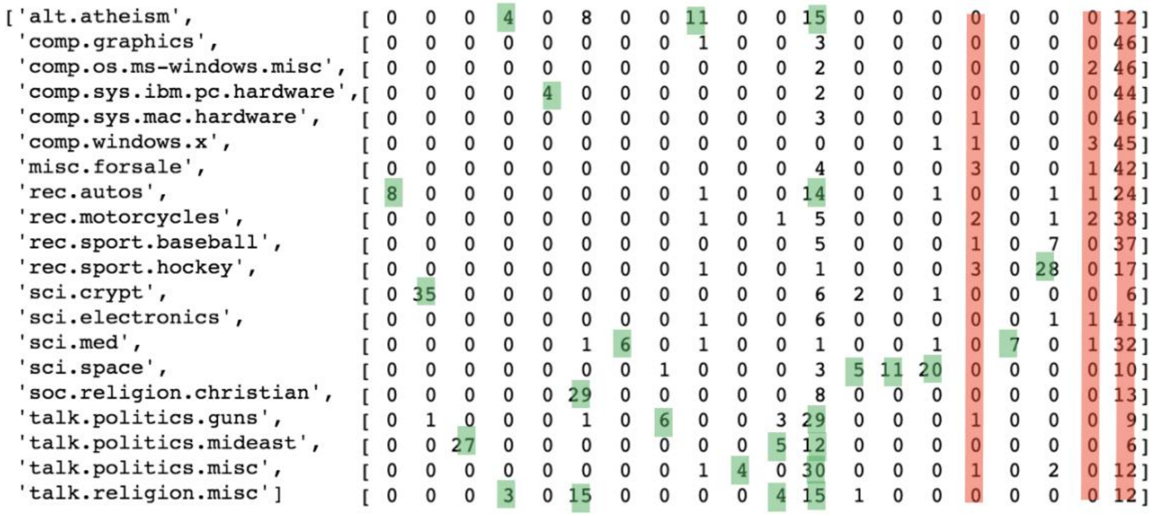
We will evaluate the models in these three scenarios,

1) 20NewsGroup with a fixed number of clusters
2) Custom Dataset with a fixed number of clusters
3) Custom Dataset with generalized threshold

5.2.4.1 20NewsGroup with fixed number of clusters

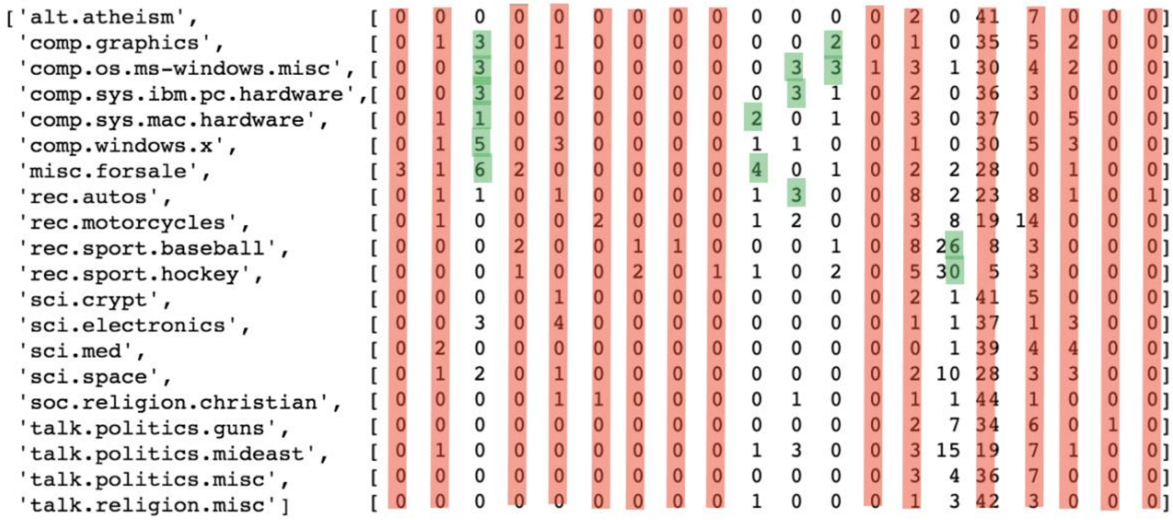
	Adjusted Rand Score	Homogeneity	Completeness	V Measure	Silhouette	Adjusted Silhouette	Loss
TF-IDF	0.060	**0.478**	0.281	**0.354**	0.008	-0.033	562
Word2Vec	0.018	0.231	0.115	0.154	**0.137**	0.030	822
USE	**0.091**	0.362	**0.313**	0.336	0.057	**0.141**	480
BERT	0.079	0.342	0.292	0.315	0.015	0.062	**389**

TF-IDF's Contingency Matrix:



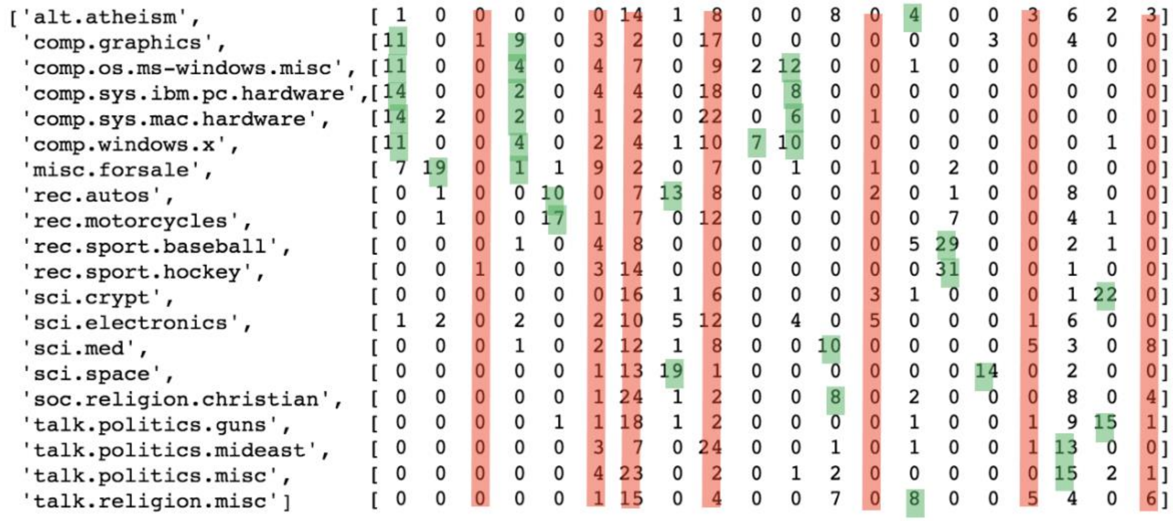
```
['alt.atheism',             [ 0  0  0  4  0  8  0  0 11  0  0 15  0  0  0  0  0  0  0 12]
 'comp.graphics',           [ 0  0  0  0  0  0  0  0  1  0  0  3  0  0  0  0  0  0  0 46]
 'comp.os.ms-windows.misc', [ 0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  2 46]
 'comp.sys.ibm.pc.hardware',[ 0  0  0  0  4  0  0  0  0  0  0  2  0  0  0  0  0  0  0 44]
 'comp.sys.mac.hardware',   [ 0  0  0  0  0  0  0  0  0  0  0  3  0  0  0  1  0  0  0 46]
 'comp.windows.x',          [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  3 45]
 'misc.forsale',            [ 0  0  0  0  0  0  0  0  0  0  0  4  0  0  0  3  0  0  1 42]
 'rec.autos',               [ 8  0  0  0  0  0  0  0  1  0  0 14  0  0  1  0  0  1  1 24]
 'rec.motorcycles',         [ 0  0  0  0  0  0  0  0  1  0  1  5  0  0  0  2  0  1  2 38]
 'rec.sport.baseball',      [ 0  0  0  0  0  0  0  0  0  0  0  5  0  0  0  1  0  7  0 37]
 'rec.sport.hockey',        [ 0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  3  0 28  0 17]
 'sci.crypt',               [ 0 35  0  0  0  0  0  0  0  0  0  6  2  0  1  0  0  0  0  6]
 'sci.electronics',         [ 0  0  0  0  0  0  0  0  1  0  0  6  0  0  0  0  0  1  1 41]
 'sci.med',                 [ 0  0  0  0  0  1  6  0  1  0  0  1  0  0  1  0  7  0  1 32]
 'sci.space',               [ 0  0  0  0  0  0  0  1  0  0  0  3  5 11 20  0  0  0  0 10]
 'soc.religion.christian',  [ 0  0  0  0  0 29  0  0  0  0  0  8  0  0  0  0  0  0  0 13]
 'talk.politics.guns',      [ 0  1  0  0  0  1  0  6  0  0  3 29  0  0  0  1  0  0  0  9]
 'talk.politics.mideast',   [ 0  0 27  0  0  0  0  0  0  0  5 12  0  0  0  0  0  0  0  6]
 'talk.politics.misc',      [ 0  0  0  0  0  0  0  0  1  4  0 30  0  0  0  1  0  2  0 12]
 'talk.religion.misc']      [ 0  0  0  3  0 15  0  0  0  0  4 15  1  0  0  0  0  0  0 12]
```

Word2Vec's Contingency Matrix:



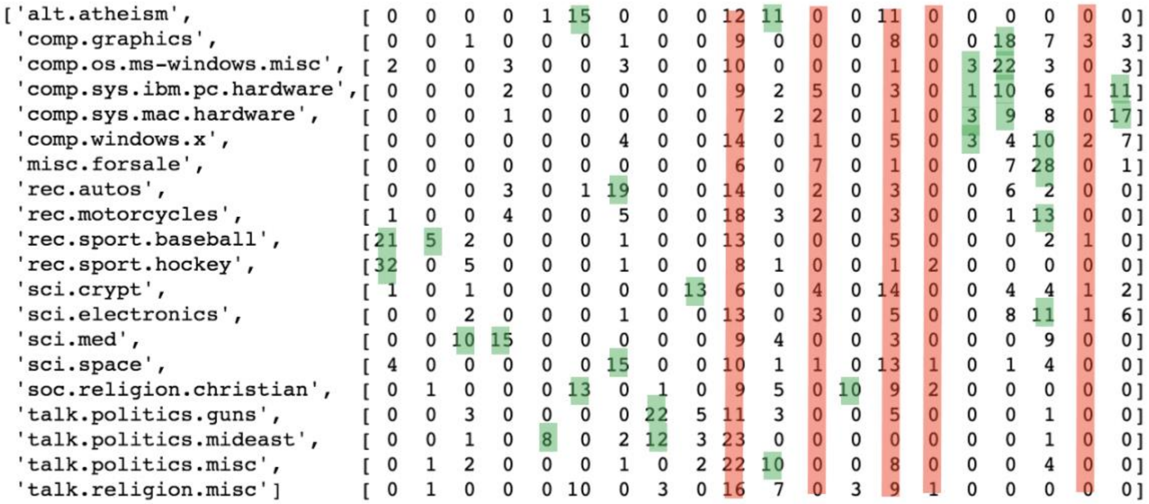
```
['alt.atheism',             [ 0  0  0  0  0  0  0  0  0  0  0  0  0  2  0 41  7  0  0  0]
 'comp.graphics',           [ 0  1  3  0  1  0  0  0  0  0  0  2  0  1  0 35  5  2  0  0]
 'comp.os.ms-windows.misc', [ 0  0  3  0  0  0  0  0  0  0  3  3  1  3  1 30  4  2  0  0]
 'comp.sys.ibm.pc.hardware',[ 0  0  3  0  2  0  0  0  0  0  3  1  0  2  0 36  3  0  0  0]
 'comp.sys.mac.hardware',   [ 0  1  1  0  0  0  0  0  0  2  0  1  0  3  0 37  0  5  0  0]
 'comp.windows.x',          [ 0  1  5  0  3  0  0  0  0  1  1  0  0  1  0 30  5  3  0  0]
 'misc.forsale',            [ 3  1  6  2  0  0  0  0  0  4  0  1  0  2  2 28  0  1  0  0]
 'rec.autos',               [ 0  1  1  0  1  0  0  0  0  1  3  0  0  8  2 23  8  1  0  1]
 'rec.motorcycles',         [ 0  1  0  0  0  2  0  0  0  1  2  0  0  3  8 19 14  0  0  0]
 'rec.sport.baseball',      [ 0  0  0  2  0  0  1  1  0  0  0  1  0  8 26  8  3  0  0  0]
 'rec.sport.hockey',        [ 0  0  0  1  0  0  2  0  1  1  0  2  0  5 30  5  3  0  0  0]
 'sci.crypt',               [ 0  0  0  0  1  0  0  0  0  0  0  0  0  2  1 41  5  0  0  0]
 'sci.electronics',         [ 0  0  3  0  4  0  0  0  0  0  0  0  0  1  1 37  1  3  0  0]
 'sci.med',                 [ 0  2  0  0  0  0  0  0  0  0  0  0  0  0  1 39  4  4  0  0]
 'sci.space',               [ 0  1  2  0  1  0  0  0  0  0  0  0  0  2 10 28  3  3  0  0]
 'soc.religion.christian',  [ 0  0  0  0  1  1  0  0  0  0  1  0  0  1  1 44  1  0  0  0]
 'talk.politics.guns',      [ 0  0  0  0  0  0  0  0  0  0  0  0  0  2  7 34  6  0  1  0]
 'talk.politics.mideast',   [ 0  1  0  0  0  0  0  0  0  1  3  0  0  3 15 19  7  1  0  0]
 'talk.politics.misc',      [ 0  0  0  0  0  0  0  0  0  0  0  0  0  3  4 36  7  0  0  0]
 'talk.religion.misc']      [ 0  0  0  0  0  0  0  0  0  1  0  0  0  1  3 42  3  0  0  0]
```

USE's Contingency Matrix:



```
['alt.atheism',             [ 1  0  0  0  0  0 14  1  8  0  0  8  0  4  0  0  3  6  2  3]
 'comp.graphics',           [11  0  1  9  0  3  2  0 17  0  0  0  0  0  0  3  0  4  0  0]
 'comp.os.ms-windows.misc', [11  0  0  4  0  4  7  0  9  2 12  0  0  1  0  0  0  0  0  0]
 'comp.sys.ibm.pc.hardware',[14  0  0  2  0  4  4  0 18  0  8  0  0  0  0  0  0  0  0  0]
 'comp.sys.mac.hardware',   [14  2  0  2  0  1  2  0 22  0  6  0  1  0  0  0  0  0  0  0]
 'comp.windows.x',          [11  0  0  4  0  2  4  1 10  7 10  0  0  0  0  0  0  0  1  0]
 'misc.forsale',            [ 7 19  0  1  1  9  2  0  7  0  1  0  1  0  2  0  0  0  0  0]
 'rec.autos',               [ 0  1  0  0 10  0  7 13  8  0  0  0  2  0  1  0  0  8  0  0]
 'rec.motorcycles',         [ 0  1  0  0 17  1  7  0 12  0  0  0  0  0  7  0  0  4  1  0]
 'rec.sport.baseball',      [ 0  0  0  1  0  4  8  0  0  0  0  0  0  5 29  0  0  2  1  0]
 'rec.sport.hockey',        [ 0  0  1  0  0  3 14  0  0  0  0  0  0  0 31  0  0  1  0  0]
 'sci.crypt',               [ 0  0  0  0  0  0 16  1  6  0  0  0  3  1  0  0  0  1 22  0]
 'sci.electronics',         [ 1  2  0  2  0  2 10  5 12  0  4  0  5  0  0  0  1  6  0  0]
 'sci.med',                 [ 0  0  0  1  0  2 12  1  8  0  0 10  0  0  0  0  5  3  0  8]
 'sci.space',               [ 0  0  0  0  0  1 13 19  1  0  0  0  0  0  0 14  0  2  0  0]
 'soc.religion.christian',  [ 0  0  0  0  0  1 24  1  2  0  0  8  0  2  0  0  0  8  0  4]
 'talk.politics.guns',      [ 0  0  0  0  1  1 18  1  2  0  0  0  0  1  0  0  1  9 15  1]
 'talk.politics.mideast',   [ 0  0  0  0  0  3  7  0 24  0  0  1  0  1  0  0  1 13  0  0]
 'talk.politics.misc',      [ 0  0  0  0  0  4 23  0  2  0  1  2  0  0  0  0  0 15  2  1]
 'talk.religion.misc']      [ 0  0  0  0  0  1 15  0  4  0  0  7  0  8  0  0  5  4  0  6]
```

BERT's Contingency Matrix:



```
['alt.atheism',             [ 0  0  0  0  1 15  0  0  0 12 11  0  0 11  0  0  0  0  0  0]
 'comp.graphics',           [ 0  0  1  0  0  0  1  0  0  9  0  0  0  8  0  0 18  7  3  3]
 'comp.os.ms-windows.misc', [ 2  0  0  3  0  0  3  0  0 10  0  0  0  1  0  3 22  3  0  3]
 'comp.sys.ibm.pc.hardware',[ 0  0  0  2  0  0  0  0  0  9  2  5  0  3  0  1 10  6  1 11]
 'comp.sys.mac.hardware',   [ 0  0  0  1  0  0  0  0  0  7  2  2  0  1  0  3  9  8  0 17]
 'comp.windows.x',          [ 0  0  0  0  0  0  4  0  0 14  0  1  0  5  0  3  4 10  2  7]
 'misc.forsale',            [ 0  0  0  0  0  0  0  0  0  6  0  7  0  1  0  0  7 28  0  1]
 'rec.autos',               [ 0  0  0  3  0  1 19  0  0 14  0  2  0  3  0  0  6  2  0  0]
 'rec.motorcycles',         [ 1  0  0  4  0  0  5  0  0 18  3  2  0  3  0  0  1 13  0  0]
 'rec.sport.baseball',      [21  5  2  0  0  0  1  0  0 13  0  0  0  5  0  0  0  2  1  0]
 'rec.sport.hockey',        [32  0  5  0  0  0  1  0  0  8  1  0  0  1  2  0  0  0  0  0]
 'sci.crypt',               [ 1  0  1  0  0  0  0  0 13  6  0  4  0 14  0  0  4  4  1  2]
 'sci.electronics',         [ 0  0  2  0  0  0  1  0  0 13  0  3  0  5  0  0  8 11  1  6]
 'sci.med',                 [ 0  0 10 15  0  0  0  0  0  9  4  0  0  3  0  0  0  9  0  0]
 'sci.space',               [ 4  0  0  0  0  0 15  0  0 10  1  1  0 13  1  0  1  4  0  0]
 'soc.religion.christian',  [ 0  1  0  0  0 13  0  1  0  9  5  0 10  9  2  0  0  0  0  0]
 'talk.politics.guns',      [ 0  0  3  0  0  0  0 22  5 11  3  0  0  5  0  0  0  1  0  0]
 'talk.politics.mideast',   [ 0  0  1  0  8  0  2 12  3 23  0  0  0  0  0  0  0  1  0  0]
 'talk.politics.misc',      [ 0  1  2  0  0  0  1  0  2 22 10  0  0  8  0  0  0  4  0  0]
 'talk.religion.misc']      [ 0  1  0  0  0 10  0  3  0 16  7  0  3  9  1  0  0  0  0  0]
```

Evaluation:

20NewsGroup is a challenging dataset that involves 20 different classes, and among these 20 classes, some classes are closely related which can form certain sub-categories. There are mainly 4 sub-categories, 1) topics related to politics / religion (politics), 2) topics related to science like medicine / space / cryptography (science), 3) topics related to recreation such as cars / sports (rec), 4) topics related to computer such as Windows / Mac (comp).

From the quantitative results we can see, Word2Vec has the worst performance and USE have the best performance if we look at the Rand Score. BERT has a close result to USE in most metrics (+- 0.02 in most) except the Silhouette score, which can conclude USE generates better isolated clusters mathematically. TF-IDF is performing well accord to the metrics as well.

In terms of the quantitative metrics, we conclude

USE > BERT > TF-IDF > Word2Vec

Looking at the results and group some of the "meaningless clusters" for the loss metric for each of the models, we can find BERT has the least meaningless clustered data, TF-IDF / Word2Vec has lost over 50% of the data in these meaningless clusters.

Looking at the distribution of the data, as we have mentioned it would be difficult for the algorithm to perform clear distinction of 20 clusters, we focus to see if the models are able to generate clusters for the sub-categories.

	TF-IDF	Word2Vec	USE	BERT
politics	Yes	No	Partial	Yes
science	Partial	No	Partial	Partial
rec	Partial	Partial	Yes	Yes
comp	No	Partial	Yes	Yes

In terms of the results evaluated subjectively by human, we conclude:

BERT > USE > TF-IDF > Word2Vec

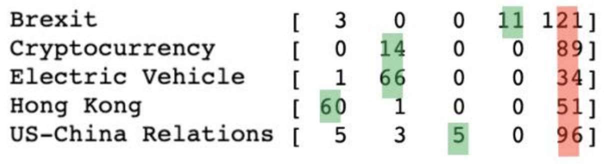
We can conclude the more advanced models (USE, BERT) is considerably better than the traditional models. Also, we can see of Word2Vec (Averaging vector) produces unexpectedly poor performance, this implies averaging vector does not work for Word2Vec's word embeddings.

5.2.4.2 Custom Dataset with a fixed number of clusters

Since 20NewsGroup's data are forum posts focused on different topics, we are more interested to see how the models differ in scenarios more similar to SG's daily use cases. Our custom dataset is designed for this purpose, therefore we perform the same evaluation setting on the custom dataset.

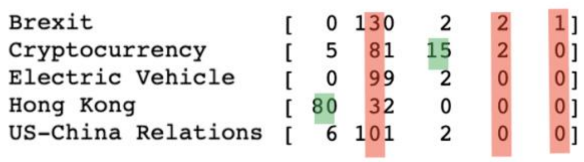
	Adjusted Rand Score	Homogeneity	Completeness	V Measure	Silhouette	Adjusted Silhouette	Loss
TF-IDF	0.128	0.415	0.236	0.301	0.006	0.141	391
Word2Vec	0.103	0.429	0.174	0.248	0.092	0.063	465
USE	0.484	0.572	0.500	0.534	**0.125**	**0.363**	**11**
BERT	**0.559**	**0.656**	**0.583**	**0.617**	0.099	0.347	23

TF-IDF



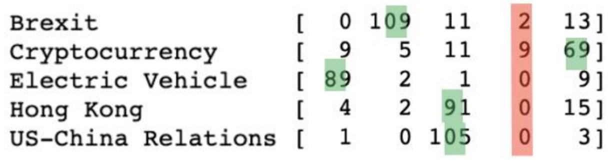
```
Brexit               [  3   0   0  11 121]
Cryptocurrency       [  0  14   0   0  89]
Electric Vehicle     [  1  66   0   0  34]
Hong Kong            [ 60   1   0   0  51]
US-China Relations   [  5   3   5   0  96]
```

Word2Vec



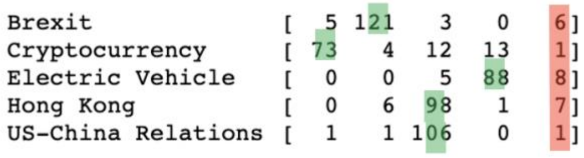
```
Brexit               [  0 130   2   2   1]
Cryptocurrency       [  5  81  15   2   0]
Electric Vehicle     [  0  99   2   0   0]
Hong Kong            [ 80  32   0   0   0]
US-China Relations   [  6 101   2   0   0]
```

USE



```
Brexit               [  0 109  11   2  13]
Cryptocurrency       [  9   5  11   9  69]
Electric Vehicle     [ 89   2   1   0   9]
Hong Kong            [  4   2  91   0  15]
US-China Relations   [  1   0 105   0   3]
```

BERT



```
Brexit               [  5 121   3   0   6]
Cryptocurrency       [ 73   4  12  13   1]
Electric Vehicle     [  0   0   5  88   8]
Hong Kong            [  0   6  98   1   7]
US-China Relations   [  1   1 106   0   1]
```

Evaluation:

Running the models on the custom dataset, it is more differentiable the strength of the more advanced models. From the quantitative results, we can see BERT secured a clear leading position, shortly after it is the USE model. Looking at the more traditional models, both TF-IDF and Word2Vec cannot produce satisfying results. We conclude,

BERT > USE > TF-IDF > Word2Vec

Looking at the clustering results, it echoes our quantitative results. The majority of the data in TF-IDF and Word2Vec goes to the "meaningless clusters", almost reaching 80% of total data, we can learn these models are unable to cluster this dataset. On the other hand, both USE and BERT have a clear distinction for each of the 5 classes, especially "Brexit", "Cryptocurrency", "Electric Vehicle". "Hong Kong" and "US-China Relations" have overlapped in both models, which we concluded is a reasonable overlapping as they carry similar semantic relations. In most of these distinctive clusters, we can spot BERT performed slightly better as we can see generally its clusters have less noise. We conclude,

BERT > USE > TF-IDF > Word2Vec

5.2.4.3 Custom Dataset with generalized threshold

Apart, we want to understand how the models perform if there is no knowledge of the target threshold. This would be more similar to the realistic scenarios. Since both evaluations have shown the transformer models (BERT, USE) led to much better results, we will only evaluate these two models with a threshold of 0.86. We referenced this threshold from Bloomberg.

	Adjusted Rand Score	Homogeneity	Completeness	V Measure	Silhouette	Adjusted Silhouette	No. of Clusters
USE	**0.321**	**0.418**	0.859	**0.562**	**0.097**	0.274	50
BERT	0.180	0.372	**0.874**	0.522	0.044	**0.280**	61

Evaluation:

We can see both models generated much more clusters than we desired using this threshold. The performance of USE looks better than BERT in the metrics, however we should stress this kind of evaluation is more to evaluate the threshold rather than the model. USE performs better with this threshold probably because the vectors generated by USE are more similar to each other than BERT, resulting in fewer clusters when cut off at the distance of 0.86. There are no better or worse for whether the vectors are more similar or less because it could be adjusted by choosing another threshold, and we have already concluded BERT performs better than USE under a fixed number of clusters which is a fair scenario. Nonetheless, this exposes a problem that is we are generating too many clusters and we would need certain techniques to reduce them or adjust the threshold.

5.2.4.4 Conclusion

From the results we ran over the two datasets, we conclude the transformer model (BERT, USE) performs the best that they are able to generate moderate to good results. TF-IDF performs poorly as expected, given the difficulty of our language task is much higher than its capacity. Word2Vec as well performed poorly, however not because of the model but because of averaging vector does not work in that way as expected.

Among BERT and USE, we decided to go with BERT for serval reasons, 1) It performed better than USE in our custom dataset task, 2) It used a more novel design which proved to have a good performance on multiple NLP benchmarks, 3) Its library provide more customization and options for us to fit our need.

There are also problems we can spot.

1) The models can only run-on general dataset: From the evaluation of 20NewsGroup we can learn, all models can only cluster the dataset in the sub-category level and are unable to perform a clear distinctive clustering on specific classes. This implies difficulty when we want to cluster news with some specific topic (Example: Clustering on news focused on certain companies)

2) We have gathered the clustering algorithm and word embedding model, however they output more than the expected number of clusters with the generalized threshold. Although most of the clusters are correctly classified sub-clusters (We can learn this from the dendrogram of custom datasets, the 61 clusters will soon conquer to 5 clusters, therefore if the 5 clusters are correctly classified, these 61 clusters will also be correctly classified), we need to find method either to merge these clusters or find a better threshold. (To be discussed in section 5.4)

5.3 Named Entity Recognition

On top of Clustering, we intend to extract the keywords, including most frequent name entities and nouns, in order to offer the readers some inductive insights about the major topics covered per cluster.

5.3.1 Library

SpaCy is a Python library for advanced Natural Language Processing Language. It has a sophisticated set of algorithms, including identifying Part-of-Speech tagging and Name Entities. It is also free and open-source, which is suitable for our project with limited resources.

SpaCy and NLTK are the two most common methods. Yet, SpaCy is considered outperforming its major counterpart, NLTK in aspects of speed performance and coverage of support (Malhotra, 2018). As a result, we presume that SpaCy library is a reasonably good library among a wide range of alternatives.

Before the keywords extractions, we use the pre-trained pipeline to tokenize each cluster first. Among 4 English tokenization pipelines, we adopt the "*en_core_web_lg*" which includes the most unique words.

5.3.2 Named Entity Extraction

A named entity is the identified name in the real world. After using a trained model to tokenize the headlines in a cluster, SpaCy can recognize various types of named entities in the text. For the sake of offering to-the-point insights about the cluster for the analysts at SG, we exclude named entity types such as date, time, percent, work of art, quantity, which are too specific and hints little about the general topics covered in the clusters. Thus only the following types are included:

PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
EVENT	Named hurricanes, battles, wars, sports events, etc.

By counting the frequency of the related named entities of the above types in each cluster, the 'Top Named Entities' will be generated as output and displayed on the web portal after this pipeline.

5.3.3 Noun Extraction

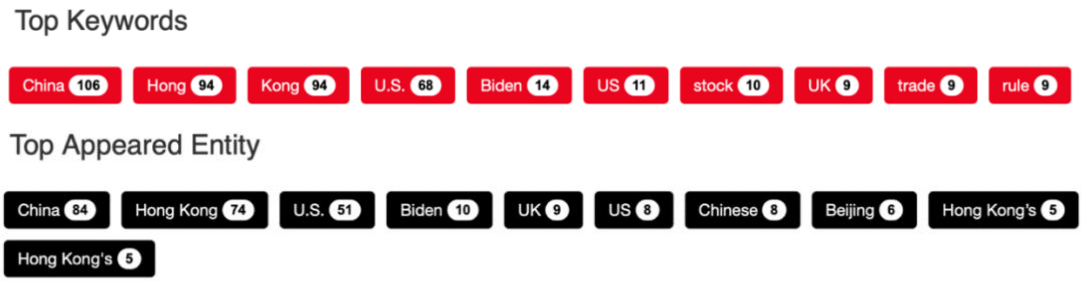
The Top Named Entities has an inherent blind spot that only named entities included by the trained pipelines can be identified. Newly coined and hot terms might not be recognized by the pre-trained pipelines, for instance, "dogecoin" as a kind of cryptocurrency. While this project does not have sufficient resources to self-train a pipeline, we decide to extract the most common nouns in each cluster as a second set of keywords to complement the lagging nature of top named entities.

SpaCy tokenization pipelines can parse and tag the part of speech of a given word in the document. Among all the parts of speech, we decide to extract only "Noun" and "Pronoun". It is presumed that the nouns and pronouns carry more unique information than other parts of speech such as adjective sand verbs and is more representative of the gist of the text. Despite some information loss, we prioritized the objective of minimizing meaningless information and avoiding wasting the time of the analysts in filtering away them.

In the current stage of development, there are some duplicated keywords generated in both Top Named Entities and Top Nouns for one cluster. To resolve the duplication, we propose to set a rule-based dictionary to map the ambiguous or duplicated outputs to a correct expression (e.g. "Hong", "Kong", "HK" to "Hong Kong"). Note that this proposed solution is not scalable and a sustainable solution should be explored, such as self-training an NLP pipeline.

5.3.4 Results Demonstration

Below top keywords extracted from Cluster 1 (size: 215 news) of the custom dataset.



Top Keywords

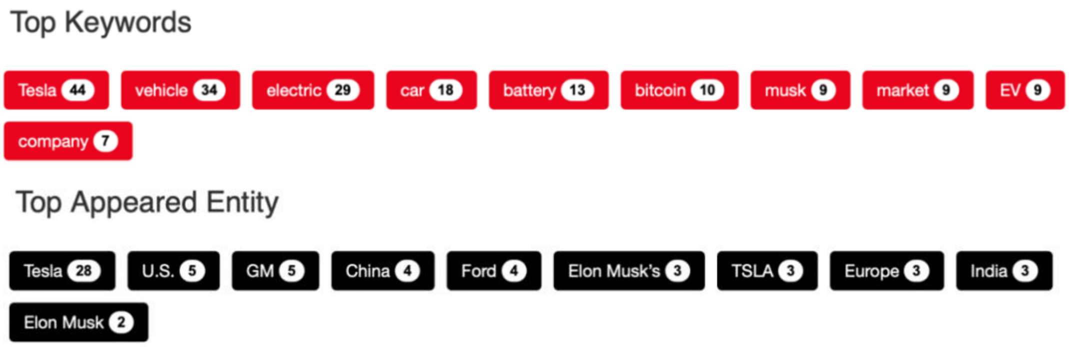
China 106 Hong 94 Kong 94 U.S. 68 Biden 14 US 11 stock 10 UK 9 trade 9 rule 9

Top Appeared Entity

China 84 Hong Kong 74 U.S. 51 Biden 10 UK 9 US 8 Chinese 8 Beijing 6 Hong Kong's 5 Hong Kong's 5

We can generalize that this cluster is associated with the relations between Hong Kong, the US and China because of the high frequency of their names. If we examine the actual headlines contained in this cluster on our web demo, we can verify that the top keywords extracted match the topics covered.

In contrast, Cluster 2 (size: 112) generates keywords related to electric vehicles with the frequent appearance of Tesla, battery and electric car.



Top Keywords

Tesla 44 vehicle 34 electric 29 car 18 battery 13 bitcoin 10 musk 9 market 9 EV 9 company 7

Top Appeared Entity

Tesla 28 U.S. 5 GM 5 China 4 Ford 4 Elon Musk's 3 TSLA 3 Europe 3 India 3 Elon Musk 2

From above, the top keywords extracted can succinctly indicate the topics centred by the news headlines contained in a given cluster. We conclude that this functionality achieves our objective of providing to-the-point and inductive insights about the content of each cluster for the reader.

We have also clusters the news and generates keywords per cluster for the UCI News Aggregator dataset. More demonstrations can be referred to our web demo (http://hkusg.herokuapp.com).

5.4 Enhancement

This section would like to enhance our clustering pipeline in addressing some of the known issues, in particular reducing the number of clusters generated by the current pipeline. We think of two approaches to solve this problem, they are 1) Merge similar clusters and 2) Use a larger threshold. We tend to use approach 1 rather than approach 2 since keep changing the threshold accord to the content is not sustainable, we could imagine in certain days there may have more diverse topics of news which we need a larger threshold and certain days there are more specific topics which we need a smaller threshold. Nonetheless, we have taken into account both approaches.

5.4.1 Merging clusters with Recursive Clustering

We referenced the approach of recursive Clustering used by Bloomberg NSTM and applied it in our development (Bambrick, 2020). The idea of recursive Clustering is simple, after generating the clusters in the first run, we apply the same pipeline again on these generated clusters. The challenge lies in, how do we convert these generated clusters into numeric values, since the input has changed from a string to an array of strings. We designed three approaches to test with,

1) Concatenation
   For each cluster, we concatenate all the news titles grouped in this cluster to form a large string, and retrieve the sentence embedding of this string from BERT. Then, we perform Clustering over the new embedding we got.

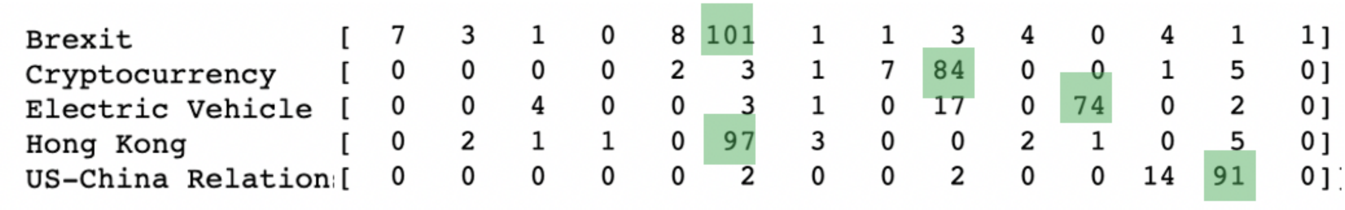
2) Averaging Vectors
   For each cluster, as we have the news titles' sentence embedding already, we take the average of all the news titles' sentence embeddings grouped in this cluster as a new sentence embedding. Then, we perform Clustering over the new embedding we got.

3) Noun Extraction
   As we discussed the NER and noun extraction in our previous section, we made use of them to formulate this approach. For each cluster, we extract the nouns from the news titles inside it, we retrieve the top 10 appeared nouns, concatenate them to form a 10 words string and retrieve the sentence embedding of this string from BERT. Then, we perform Clustering over the new embedding we got.

5.4.1.1 Results evaluation

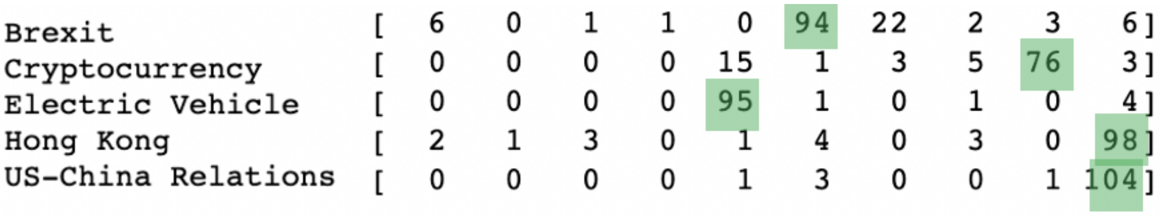
	Adjusted Rand Score	Homogeneity	Completeness	V Measure	No. of Clusters
Concatenation	0.506	0.550	0.610	0.578	14
Averaging Vectors	0.537	**0.610**	0.613	0.612	**10**
Noun Extraction	**0.614**	0.578	**0.697**	**0.632**	**10**

Concatenation's Contingency Matrix:



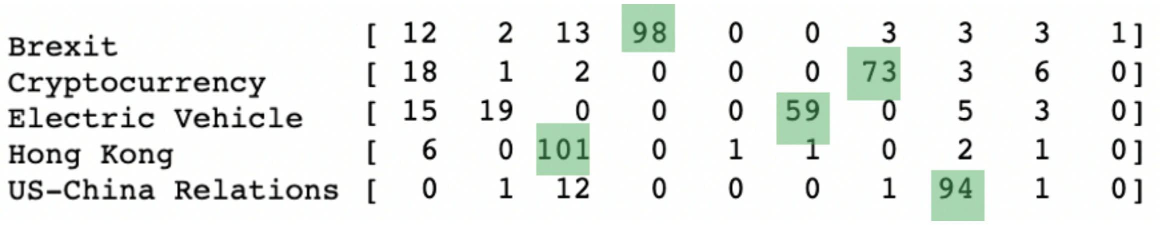
```
Brexit            [  7   3   1   0   8 101   1   1   3   4   0   4   1   1]
Cryptocurrency    [  0   0   0   0   2   3   1   7  84   0   0   1   5   0]
Electric Vehicle  [  0   0   4   0   0   3   1   0  17   0  74   0   2   0]
Hong Kong         [  0   2   1   1   0  97   3   0   0   2   1   0   5   0]
US-China Relation:[  0   0   0   0   0   2   0   0   2   0   0  14  91   0]:
```

Averaging Vector's Contingency Matrix:



```
Brexit              [  6   0   1   1   0  94  22   2   3   6]
Cryptocurrency      [  0   0   0   0  15   1   3   5  76   3]
Electric Vehicle    [  0   0   0   0  95   1   0   1   0   4]
Hong Kong           [  2   1   3   0   1   4   0   3   0  98]
US-China Relations  [  0   0   0   0   1   3   0   0   1 104]
```

Noun Extraction's Contingency Matrix:



```
Brexit             [ 12   2  13  98   0   0   3   3   3   1]
Cryptocurrency     [ 18   1   2   0   0   0  73   3   6   0]
Electric Vehicle   [ 15  19   0   0   0  59   0   5   3   0]
Hong Kong          [  6   0 101   0   1   1   0   2   1   0]
US-China Relations [  0   1  12   0   0   0   1  94   1   0]
```

Evaluation:
We have run the three approaches with BERT as their sentence embedding. It is great to see the resulting clusters have largely reduced from 61 to ~10. We will discuss each approach's performance one by one.

Concatenation has the worst performance overall, in terms of the quantitative metrics and number of clusters reduced. Looking at the data distribution, we find "Hong Kong" overlaps with "Brexit" instead of "US-China Relations", which we conclude is an unreasonable overalap. On the theoretical side, there is also concern that we cannot ensure the size of the concatenated string. This could cause two problems, 1) Unpredictable long string might cause performance issue (e.g. memory exception / long processing time), 2) Unable to ensure BERT can capture the meaning of a long string. Given these uncertainties and performance, we abandoned this approach.

Averaging Vector generates the closest result to our previous result from Section 5.2.1.6 BERT, both in quantitative metrics and matrix. We can see unique classes representing different classes and "Hong Kong" reasonably overlaps with "US-China Relations". The advantage of this approach also comes with its ease of use, it is very easy to implement and free of performance issues which provide very safe control, furthermore, this approach is also theoretically reasonable to use. We have chosen this as our final approach by its acceptable performance and stable implementation.

Noun Extraction produced the best performance. It even performed better than the result in Section 5.2.1.6 BERT in terms of the quantitative metrics. Looking at its data distribution, we can see for the first time the algorithm able to distinguish between "Hong Kong" and "US-China Relations" and produced 5 distinctive clusters for the 5 classes. However, we see 2 limitations on its theoretical side.

1) Difficult to ensure the sequence of the nouns. The sequence matters a lot in the BERT model while it is difficult to ensure multi-word nouns in its sequence.

2) Difficult to reflect the weight of nouns. Although we have already selected the top 10, there could be cases the weight ratio are very extreme in these 10 words (ex. 100 appearancees of "China" but only 1 appearance of "Iran"), results merging two non-related clusters if they represent different news.

Albeit the outstanding performance it produced, these limitations could be critical and contradict the result if not handled correctly. We therefore recommend further study some methods to overcome these limitations before deploying this approach.

5.4.1.2 Threshold finalization

We shall able to finalize the threshold after selecting the recursive approach. To choose the best threshold, we created 10 binary classification tasks from the 5 classes and tested with threshold ranging from 0.6 to 1.2 increment by 0.05. The adjusted rand scores of the results are displayed.

Threshold	**Brexit v Crypto**	**Brexit v EV**	**Brexit v HK**	**Brexit v US-China**	**Crypto v EV**	**Crypto v HK**	**Crypto v US-China**	**EV v HK**	**EV v US-China**	**HK v US-China**	**Average**
0.6	0.087	0.088	0.096	0.126	0.058	0.088	0.127	0.091	0.112	0.108	0.098
0.65	0.134	0.149	0.159	0.211	0.105	0.332	0.291	0.224	0.167	0.306	0.208
0.7	0.233	0.328	0.402	0.390	0.223	0.409	0.450	0.347	0.421	0.513	0.372
0.75	0.504	0.431	0.430	0.581	0.465	0.457	0.634	0.462	0.675	0.644	0.528
0.8	0.589	0.648	0.553	0.696	0.594	0.773	0.764	0.811	0.804	0.820	0.705
0.85	0.704	0.735	**0.679**	0.716	**0.629**	0.821	0.882	**0.889**	**0.943**	**0.868**	**0.787**
0.9	**0.845**	**0.924**	0.006	**0.774**	0	**0.890**	**0.907**	-0.001	0	0.000	0.435
0.95	0	-0.002	0	-0.002	0	-0.002	0	-0.001	0	0.000	-0.001
1	0	0	0	0	0	0	0	0	0	0	0
1.05	0	0	0	0	0	0	0	0	0	0	0
1.1	0	0	0	0	0	0	0	0	0	0	0
1.15	0	0	0	0	0	0	0	0	0	0	0
1.2	0	0	0	0	0	0	0	0	0	0	0

As we can see from the results, 0.85 – 0.9 have the best result on average. 0.85 will be an ideal threshold for the pipeline with recursive Clustering.

5.4.2 Employ a larger threshold

A simpler approach to the above searching method is using a larger threshold, i.e. we can simply select the threshold that outputs 5 clusters. To better understand the ideal threshold with this approach, we created a similar threshold test. The adjusted rand scores of the results are displayed.

Threshold	Brexit v Crypto	Brexit v EV	Brexit v HK	Brexit v US-China	Crypto v EV	Crypto v HK	Crypto v US-China	EV v HK	EV v US-China	HK v US-China	Average
0.6	0.011	0.010	0.012	0.014	0.012	0.015	0.017	0.014	0.016	0.017	0.014
0.65	0.018	0.017	0.020	0.024	0.019	0.025	0.030	0.022	0.026	0.028	0.023
0.7	0.023	0.027	0.030	0.032	0.031	0.035	0.041	0.037	0.038	0.043	0.034
0.75	0.036	0.034	0.044	0.065	0.044	0.063	0.055	0.064	0.058	0.062	0.052
0.8	0.053	0.048	0.081	0.085	0.054	0.091	0.097	0.111	0.082	0.112	0.081
0.85	0.095	0.090	0.140	0.122	0.079	0.127	0.140	0.135	0.122	0.146	0.120
0.9	0.133	0.136	0.259	0.260	0.133	0.236	0.277	0.207	0.216	0.327	0.218
0.95	0.220	0.267	0.291	0.344	0.225	0.471	0.376	0.470	0.398	0.487	0.355
1	0.518	0.692	0.394	0.501	0.400	0.503	0.609	0.526	0.863	0.548	0.555
1.05	0.683	0.697	0.495	0.678	0.414	**0.506**	**0.907**	0.739	0.863	0.605	0.659
1.1	**0.793**	**0.879**	**0.590**	0.775	**0.538**	0.479	**0.907**	**0.820**	**0.943**	**0.640**	**0.737**
1.15	0	0	0	**0.904**	**0.538**	0.314	0	0	0	0	0.176
1.2	0	0	0	0	0	0	0	0	0	0	0

From this we can learn, if we do not use the recursive technique but simply employ a larger threshold, using 1.1 as the threshold can also yield similar results. 1.1 is also the threshold to output 5 clusters from the custom dataset.

5.4.3 Discussion of the two approaches

In this part, we would like to address the question: Reason to implement the recursive Clustering (RC) over a larger threshold (LT). Our rationale is threefold:

1) Recursive Clustering has a better and stabler performance
   From the above results we can see, RC scored 0.79 comparing to LT's 0.74 on average. Maximum and minimum of RC are 0.94 and 0.63 respectively, comparing to LT's 0.94 and 0.48, RC tends to produce more stable results.

2) Recursive Clustering allows more customization
   A reason RC may be more favorable is we can do different customizations on it, for example, the three RC approaches we have tested previously, the threshold to be used in the second Clustering, how many times of recursion to be done. We see quite a lot of potentials for us to experiment, some may lead to better results (Noun extraction for example). LT, on the other hand, offers only one parameter to fine-tune, which is much less flexible.

3) Recursive Clustering could be extended to more specific datasets in the future
Based on the customization capability RC offers, we see the potential that we can use RC to solve the pain points of the general/specific dataset we addressed in Section 5.2.4.4. The idea is simple, RC performs the first Clustering using the threshold (0.86), which generates a result of more fragmented thresholds (Result 1). After performing the second Clustering, and we found there is only one cluster left (Result 2), it implies this is a quite specific dataset, we can then revert the returning result to Result 1. In this way, we are able to target both general and specific datasets with the same algorithm, we will leave this for further study.

For these three reasons, we conclude choosing recursive Clustering can yield a better result than setting a default threshold.

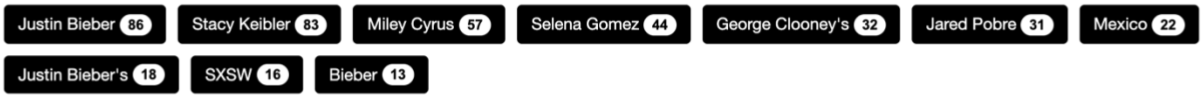
5.5 Conclusion

Here we listed all the parameters and decisions we have made for the finalized pipeline.

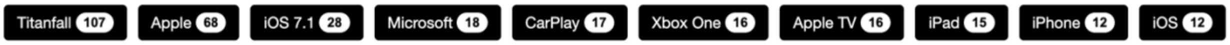
Word Embedding		1st Clustering		2nd Clustering		Name Entity Recognition	Evaluation
Model	BERT	Model	Hierarchical	Same setting as 1st		Named Entity Extraction, Noun Extraction	Rand Score, Entropy, Silhouette, Loss, Contingency Matrix
		Linkage	Complete				
		Metric	Cosine Similarity	Approach	Averaging Vector		
		Threshold	0.85				

We should stress the above setting are specifically designed for clustering general news dataset (expected to cluster from multiple topics instead of specific topics), which is the goal of our project task.

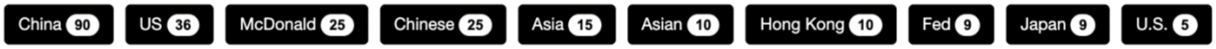
To avoid parameters overfit the custom dataset, we ran this pipeline on the UCI news aggregator dataset on a daily basis. The results can be found here (http://hkusg.herokuapp.com/uci). We are able to see the algorithm successfully clusters news into entertainment, technology, business, political news on different dates, which largely assisted users to understand what happened from an unseen bracket of news.



Example named entity extraction of Entertainment-related Cluster in “2014-03-11@UCI Dataset”



Example named entity extraction of Technology-related Cluster in “2014-03-11@UCI Dataset”



Example named entity extraction of Business-related Cluster in “2014-03-11@UCI Dataset”

From the results we evaluated with the custom dataset and the test on the UCI news aggregator dataset, we are confident this finalized pipeline can execute our project goal sufficiently well.

6 Integration and Architecture

6.1 Architecture

Our system architecture consists of four parts and the flow of data is as follows.

Newspaper API → Pipeline → Database → Web Server

The program is written in Python.

6.2 News Scrapping

Our client owns subscription accounts of various newspapers, for instance, Financial Times, Reuters, Bloomberg, Risk.net. APIs are provided by these companies. Titles, sources, URL and publish dates of news are extracted and exported in JSON data structure. In API, a time filter is set to collect news within a specific timeframe.

6.3 Pipeline

News data is imported into the pipeline. Word embedding is the first to run and is followed by clustering and named entity recognition (NER). News in the same cluster is aggregated into a JSON document. The information of cluster is recorded, e.g., the total number of articles, date. News data like articles and URLs are stored as a list. Entities generated by NER are also kept. This program is set to run Clustering in a daily manner. Its' output is saved in the database.

6.4 Database

The output is stored in the database so that the program does not need to run in real-time. This can lower the retrieval time for accessing the result and enhance the user experience.

MongoDB is chosen, and the reasons are as follows. First, MongoDB is NoSQL database, which provides less rigid document schemas. Changes and modifications are made frequently in the development stage, a NoSQL prevents developers from spending time changing table format and setting from time to time.

In addition, MongoDB stores data in JSON format, which can easily collaborate with our program. In our program, JSON format is used in newspapers' APIs and pipeline output.

6.5 Web Server

In this part, the data transfer between the frontend & backend will be mentioned.

6.5.1 HTML & Bootstrap

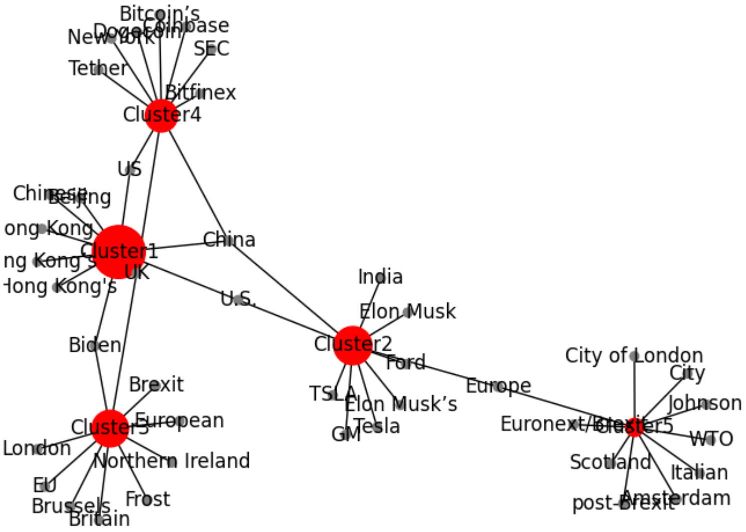
Bootstrap is a commonly used website building framework. It saves developers time by providing ready-to-use components of widely used functions. On our web, buttons, filter and dropdowns are built on it. Besides, Bootstrap has a clear grid system to enhance website layout management. In this project, Bootstrap 4.0 is used. To set up Bootstrap, multiple CSS and JavaScript stylesheets are pasted into the header part of the base HTML. The main function of the website is inside the index HTML.

6.5.2 Flask API

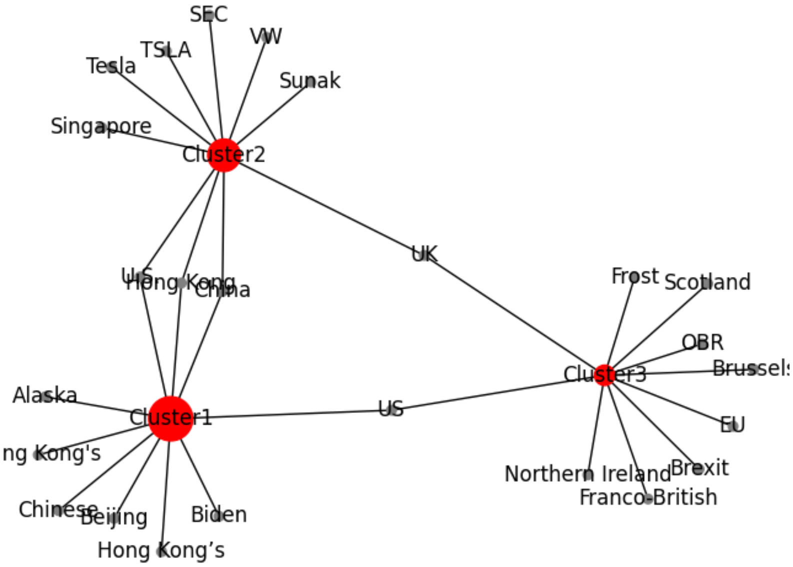
To facilitate the data transfer between MongoDB and the website, Python web API – Flask 1.1.2 is used. Flask can receive users' requests from the frontend and post website content in the frontend.

6.5.3 Network Visualization Graph

A network visualization graph is used to present the clustering result. With a graph, users do not need to look into the whole webpage. The graph is generated by Python libraries - NetworkX 2.6 and Matplotlib 3.4.1. Cluster data are obtained from MongoDB. In the graph, each cluster node is connected to a noun entity that belongs to their clusters. The node size represents the number of articles in the cluster. Some noun entities are shared by multiple clusters, which implies their themes are related in some aspects. The closer the distance between two cluster nodes, the closer relationship they have. Each clustering result has its customized graphs and is stored in the webserver.



Network Visualization Graph for all data



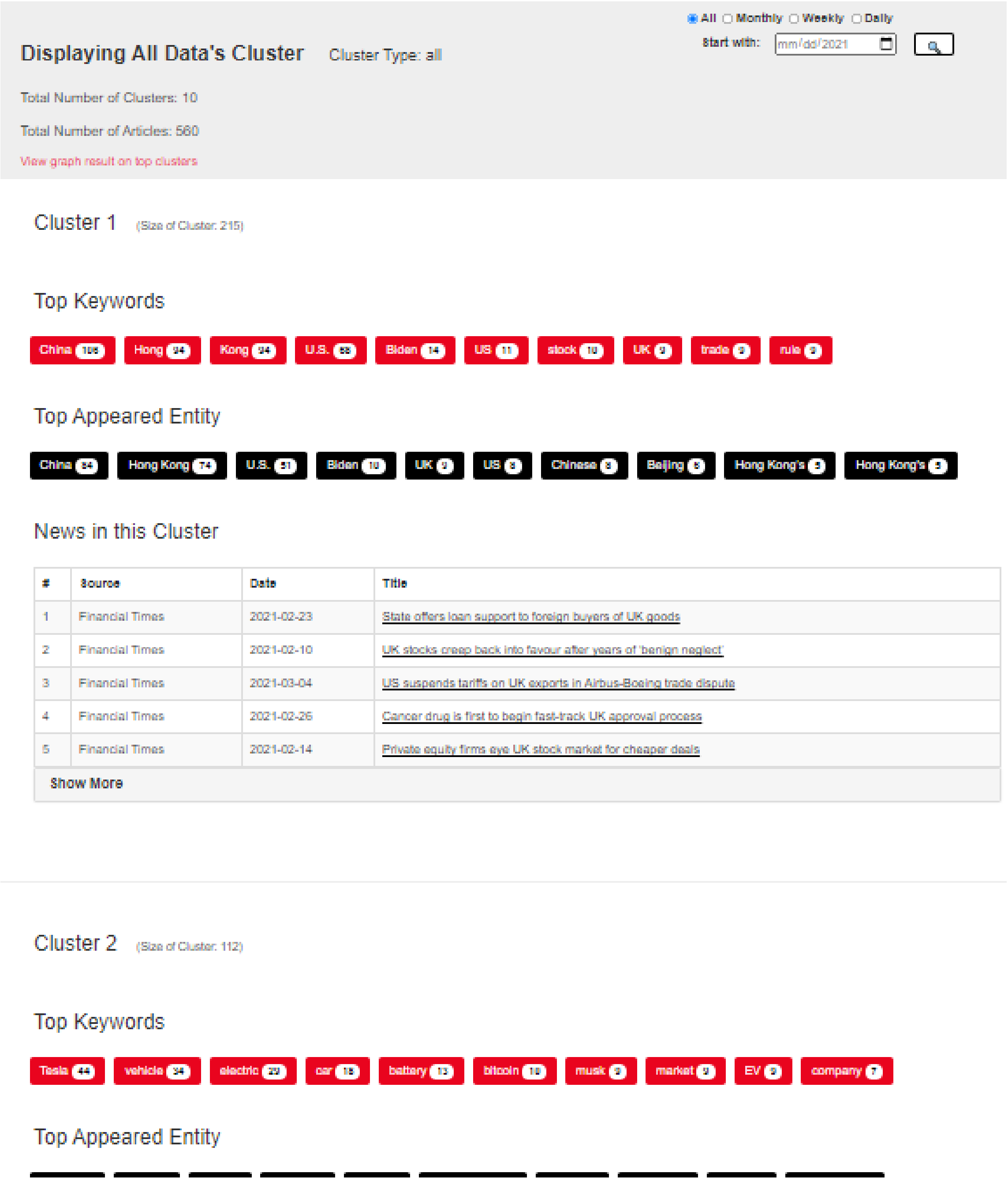
Network Visualization Graph for March only data

6.6 Website Features

To access the website, You may access it at http://hkusg.herokuapp.com/.

6.6.1 Display Clustering Result

The clustering result is displayed in two parts – the result and each cluster. At the top, cluster type, the total number of clusters, total number of articles and network graph are shown. At the body, clusters are displayed one by one. Clusters are sorted ascendingly by their sizes. Keywords and news articles under each cluster are shown. By clicking "Show More", the full list of news will be displayed. To read the full content of the news, users can click the article and redirect users to the news website.



● All ○ Monthly ○ Weekly ○ Daily

Start with: mm/dd/2021

Displaying All Data's Cluster Cluster Type: all

Total Number of Clusters: 10

Total Number of Articles: 560

View graph result on top clusters

Cluster 1 (Size of Cluster: 215)

Top Keywords

China 108 Hong 94 Kong 94 U.S. 68 Biden 14 US 11 stock 10 UK 9 trade 9 rule 9

Top Appeared Entity

China 84 Hong Kong 74 U.S. 51 Biden 10 UK 9 US 8 Chinese 8 Beijing 6 Hong Kong's 5 Hong Kong's 5

News in this Cluster

#	Source	Date	Title
1	Financial Times	2021-02-23	State offers loan support to foreign buyers of UK goods
2	Financial Times	2021-02-10	UK stocks creep back into favour after years of 'benign neglect'
3	Financial Times	2021-03-04	US suspends tariffs on UK exports in Airbus-Boeing trade dispute
4	Financial Times	2021-02-26	Cancer drug is first to begin fast-track UK approval process
5	Financial Times	2021-02-14	Private equity firms eye UK stock market for cheaper deals

Show More

Cluster 2 (Size of Cluster: 112)

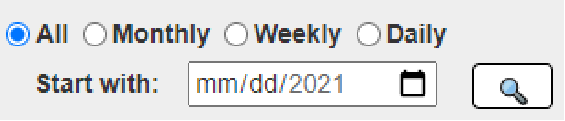
Top Keywords

Tesla 44 vehicle 34 electric 22 car 18 battery 13 bitcoin 10 musk 9 market 9 EV 9 company 7

Top Appeared Entity

6.6.2 Select Clustering Result

The clustering result of a specific date can be found by searching. Also, users can choose the timeframe of the cluster by clicking the radio button. For example, choosing "Monthly" will cluster all news within the month that the users selected.



● All ○ Monthly ○ Weekly ○ Daily

Start with: mm/dd/2021

6.7 System Integration

Our client is currently developing an NLP application and the infrastructure is similar. Our solution can be integrated into the client's infrastructure smoothly. For the named entity recognition, Clustering and word embedding, SG can integrate our suggested models if it fits SG's needs. For the database, SG is currently using MongoDB. SG can create a new folder by applying our suggested structure. For the website visualization, SG can reference our suggested features and design.

7 Future Developments

7.1 User feedbacks

From our final presentation of the prototype to the client, we received general feedback from them. They described the project as interesting and good quality work, which provides value for their data analytical tasks, and also the capacity for making information technology work as a quickly rebuilding tool. Some of the methodologies the project used are already adopted in their daily operation of their new digest system. In a more specific view, given the complicatedness of the clustering result, they hope to have a more user-friendly presentation method of result for the effective use of analysts. In general, they appreciate the interactive discussion with us during the project, especially when they could gather and understand the idea from the younger generation. They are satisfied with our progress and hope to continue collaborating with the university for more projects with students.

7.2 Future Plans

For the future enhancement of our project, we hope to work on more transparent visualisation and an expansion of the scope of Clustering. For visualisation, as mentioned by our client, the current dashboard is too simple for a lucid illustration for the result from our complicated clustering methodology, for instance, the network visualisation graph demonstrated in section 6.5.3. In our future prospect, we wish to create an interactive and convenient dashboard for users to easily understand the latest market news. Users can investigate the information they gathered from the search result through interactive graphics, from a general to specific view through their movements on devices, similar to the dashboards generated by the data visualisation software Tableau. In addition, we will be providing the most popular and most representative result by filter function to users, for that they can easily understand the importance and relationship of different clusters. All the functions will be customized for SG, by collaboration with SG analysts.

The scope of Clustering is another issue that we wish to make improvements on . Given that the current methodology mainly focuses on identifying the general topics of news, the result generated is still too broad for qualitative and penetrative analysis. In light of this, we would like to further enhance our clustering methodology, with more specific perspectives on the dataset. The overall result would be akin to Bloomberg's NSTM. It can generate results by classifying and summarizing specific issues as a cluster under the general clustered keyword. Nevertheless, real-time Clustering and the dynamic threshold would be a great challenge for us, with the limitation of time and resources as another concerns.

Overall, as a future prospect, we wish to produce a more comprehensive and penetrative new digest system through the above two perspectives, for creating an additional value to our client.

8 Management Issues

Throughout the project life, we are pleased to have invaluable and continuous feedback from the representatives of SG – from setting out the scope of the project, presenting the initial findings on different algorithms, to conducting final tweaking – we constantly test, evaluate, review, and improve the logic and design of the project deliverables. In particular, we made changes to the system based on the user feedback provided by our client. It is why we adopted agile methodology as the norm as we encourage continuous iteration of development, testing and rapid response to user feedback. Nonetheless, we have identified and mitigated a few project management issues.

8.1 Expectation Management

During the early stage of the project, the mutual expectation between the client and the project group tended to be ambiguous. For example, when we first raised the issue of unable to scrape the news content due to subscription restriction, our client offered to give us access to their internal API and subscription accounts, subject to the approval of SG's senior officier. As time goes by, in order not to cause project delay, we decided to move to the contingency plan of extracting articles headlines manually on Google News, a platform that does not require subscriptions. Fortunately, we were still able to use the Python scraping model for a few news platforms. The time taken to prepare the custom dataset was inevitably lengthened, but we still managed to have a quality dataset with reasonable difficulty. As far as the subscription issue is concerned, we mitigated the problem by adopting a contingency plan. Expectation management is the lesson to learn here – both the project group and our client should have an unambiguous understanding of what and when to expect. It requires continuous, honest, and open communication on what can be realistically achieved. In particular, the access to internal API would inevitably trigger the SG's confidentiality policies. We are pleased that such issue was eventually resolved.

8.2 Scope Changes

During the project life, we have encountered scope change. Initially, we proposed to evaluate different models for embedding and Clustering. We examined and identified 3 solutions for Clustering and 6 for sentence embedding. In addition to these algorithms, we have decided to adopt 8 metrics for evaluation purpose. A problem then arises – there would be a total of 3*6*8 = 144 parameters if each combination of Clustering and embedding method is evaluated. It would create much confusion and distort the focus of the project. Therefore, after evaluating different clustering methods, we decided only to adopt hierarchical Clustering due to its ease of implementation and computability with cosine metrics.

Scope change, as minor as it can be, is always a management issue in a project. The most important thing is to understand how it happened – is it caused by poorly defined scope? Or change of client's requirements? Then, it is for the project manager to manage the scope change. For this project, the scope change was inevitable – it was because we underestimated the variety of algorithms available. Fortunately, it did not bring many adverse impacts on the project. As project managers, we explained to the client about the scope change and delineated the updated project scope during the bi-weekly meeting. We are pleased that our client was satisfied and had given us invaluable feedback.

9 Conclusion

To conclude, the group has demonstrated a proof-of-concept for the SG news digest system in various dimensions listed as follows:

- Custom and non-custom news datasets were prepared;
- Different clustering algorithms were compared, and Hierarchical Clustering was adopted for its ease of implementation;
- Eight evaluation metrics were examined and deployed;
- Various embedding models were evaluated where pretrained models were selected for the purpose of implementation;
- Clustering results with custom dataset and 20NewsGroup were evaluated with (i) fixed number of clusters, and (ii) generalized threshold. It was concluded that the transformer models (BERT, USE) have the best performance;
- Named entities and nouns extraction were performed on top of Clustering;
- Recursive Clustering was performed where the three approaches (with BERT as embedding) were evaluated and it was concluded that a recursive clustering over a larger threshold is the best to implement; and
- A website was built where the clustering results, network graphs and time filtering are demonstrated.

The key takeaway is that the project group ***has successfully clustered news titles in a general manner without training***. With that being said, the parameters are specifically tuned for news headlines; the use case is targeted to cluster news with multiple topics, instead of specific topics; and above all, as long as the news dataset is in English, our algorithm does not require extra continuous training.

We genuinely believe that our project could add values to SG in two ways:

- Firstly, serving as the first line of analysis in the myriad of news for SG's analysts to systematically understand the market behaviour and detect anomalies; and

- Secondly, providing insights to any future NLP applications of various models, e.g. the recursive clustering approach.

As the project is approaching the end, the group would like to express its deepest gratitude to the Risk Management (Market Risk) of Société Générale (Asia) for the opportunity to conduct NLP studies. It has been an immense privilege to work with the sophisticated talents in SG and the group has learned a lot from them – from hard skills like NLP algorithms to soft skills like project management, we are incredibly grateful for their continuous support and guidance. The group has been excited to apply the theories learned in the classroom and produce a proof-of-concept for SG's news digest system. In the future, we look forward to exploring further developments on this project.

10 References

Bambrick, J., Xu, M., Almonte, A., Malioutov, I., Perarnau, G., Selo, V., & Chan, I. C. (2020). NSTM: Real-time QUERY-DRIVEN News OVERVIEW composition at Bloomberg. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations.* Retrieved from https://arxiv.org/abs/2006.01117

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B., Kurzweil, R. (2018). Universal Sentence Encoder. Retrieved from https://arxiv.org/pdf/1803.11175.pdf

Devlin, J., Chang, M., Lee, K., Toutanova, K. (2019). BERT :Pre-training of Deep Bidirectional Transformers for Language Understanding. Retrieved from https://arxiv.org/pdf/1810.04805.pdf

Malhotra, A. (2018). Introduction to Libraries of NLP in Python — NLTK vs. spaCy. Medium. Retrieved from https://medium.com/@akankshamalhotra24/introduction-to-libraries-of-nlp-in-python-nltk-vs-spacy-42d7b2f128f2

Miao, Y., Yu, L. and Blunsom, P. (2016). Neural Variational Inference for Text Processing. Retrieved from https://arxiv.org/pdf/1511.06038.pdf

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean J. (2013). Distributed Representations of Words and Phrases and their Compositionality. Retrieved from https://papers.nips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

Mikolov, T. and Le, Q. (2014). Distributed Representations of Sentences and Documents. Retrieved from https://arxiv.org/pdf/1405.4053.pdf

Müllner, D. (2013). fastcluster: Fast hierarchical clustering routines for R and Python. Retrieved from http://danifold.net/fastcluster.html?section=1

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Retrieved from https://arxiv.org/pdf/1908.10084.pdf

Rezaeinia, S., Ghodsi, A. and Rahmani, R. (2017). Improving the Accuracy of Pre-trained Word Embeddings for Sentiment Analysis. *University of Waterloo & University of Tehran.* Retrieved from https://arxiv.org/pdf/1711.08609.pdf

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65. doi:10.1016/0377-0427(87)90125-7

Scikit-Learn. (n.d.). 2.3. Clustering¶. Retrieved May 07, 2021, from https://scikit-learn.org/stable/modules/clustering.html

Stackoverflow (2015). DBSCAN error with cosine metric in python. Retrieved from https://stackoverflow.com/questions/32745541/dbscan-error-with-cosine-metric-in-python

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L. and Polosukhin, I. (2017.) Attention Is All You Need. Retrieved from https://arxiv.org/pdf/1706.03762.pdf

11 Appendix

11.1 Code Files

Github repository: https://github.com/tonywcs/hku_poc

11.2 Division of Labor

Name	Role	Tasks
Ng Long Yin, Felix	Software Engineer	• System Analysis • Development (Infrastructure)
Ng Yee Nok, Enoch	Team Leader Business Analyst	• Project Planning • Business Analysis • Documentation
Pang Tsz Fung, Sam	Software Engineer Scrum Master	• Research and development (Clustering)
Tsoi Hiu Lam, Riley	Business Analyst	• Project Planning • Business Analysis • Documentation
Wong Cheuk Shing, Tony	Software Engineer	• Research and development (Clustering)
Wong Kit Long, Marcus	Software Engineer	• Development (Website) • Testing

11.3 Team Member Information

Last Name	First Name	Preferred Name	Email	Curriculum & Year
Ng	Long Yin	Felix	longyin5@connect.hku.hk	BA IV
Ng	Yee Nok	Enoch	enochnyn@connect.hku.hk	BBA(Law)&LLB IV
Pang	Tsz Fung	Sam	u3536693@connect.hku.hk	BBA(Law)&LLB IV
Tsoi	Hiu Lam	Riley	riley311@connect.hku.hk	BBA(Law)&LLB IV
Wong	Cheuk Shing	Tony	realtony@connect.hku.hk	BBA(IS) IV
Wong	Kit Long	Marcus	u3549024@connect.hku.hk	BBA(Econ) IV

11.4 Presentation Videos

Prototype Presentation to HKU: https://youtu.be/VA5VLFTtOZ0

Substantive Presentation to the Risk Management (Market Risk) of SG (Asia): https://drive.google.com/file/d/1azYFrpxHyLdqWXctGYpZXrGcCK8nXh3J/view?usp=sharing
(Attendees included the deputy head of Risk Management Asia and others)

Final Presentation to HKU: https://youtu.be/FMrZJ6w0jbk

11.5 Photo Gallery

Apart from the online Zoom meetings, the project group has held bi-weekly face-to-face meetings with clients. The group has adopted a user-oriented approach in which user feedbacks are promptly engaged and reflected in various stages of the project.

Note: As the pandemic evolves throughout the project life, we have enforced stringent infectious control measures for the meetings – including mandatory mask w earing, social distancing, prohibition of food and drinks and temperature checks. Attendees with symptoms would be prohibited to join.

From left to right: Marcus, Tony, Ms.Yiming Fu, Mr. Ray Wong, Enoch, Sam, Riley, Felix

END OF REPORT