Textual Tweets Clustering based on User's Interests: A Comparative Study of K-Means, DBSCAN, and Hierarchical Agglomerative Clustering Schemes

Xu Han
emvx@sina.com
Science and Engineering Faculty
Queensland University of Technology
2, George Street, Brisbane, 4001, QLD, Australia

***Abstract*—Online Social Network (OSN) has become an important medium connected most people worldwide. It offers services enabling people to freely communicate, share recent news, ongoing activities, and various topics at anytime and anywhere. Twitter is a social media giant, providing the microblogging services to 145 million active users, and approximately generating 500 million tweets daily. This large-scale data stimulates many researchers on tweets mining for improving personalized recommendation system. Supervised classification is widely acknowledged that the technique can enhance recommendation quality. However, this method is heavily reliant on human transcription and annotation. With time passes, topic drift enables this work to be endless and tedious. The unsupervised clustering methods, such as K-Means, DBSCAN and Hierarchical Agglomerative Clustering, are proposed as the candidates of a novel way to be efficient in tweets clustering based on user's preferences, without the need for manual annotation. However, few experiments donate to valid the approach on real scenarios. This study will design a model combined with the three schemes and TF-IDF, for validating this idea. Silhouette Coefficient applies to determine model performance. Our experimental result indicates the three schemes are not applicable to tweets clustering currently. The reasons are the failure of creating a high-density matrix by TF-IDF and the lack of handling informal expressions or misspelling to pre-processing, which significantly impedes models on tweets clustering. Our suggestion to improvement is adopting doc2Vec to replace TF-IDF for tweets vectorization and embedding Bing Spell Check to pre-processing step.**

Keywords—TF-IDF, Unlabeled Dataset, DBSCAN, K-Means, Hierarchical Agglomerative Clustering, Tweets clustering, Similarity Measure, Silhouette Coefficient

I. INTRODUCTION

In the information age, as the explosion of social media and microblogging services, it has become one of the important social platforms, that enables people to share recent news, ongoing activities, and different topics. Twitter is an online social network (OSN), supporting approximate 145 million active users to create and share 500 million tweets, called User-Generated Content (UGC), per day [22]. The big social data stimulates active research on improving personalized recommendation systems, which is based on filtering UGC to deduce user's interests [24]. UGC is an indicator that implicitly reflects a topic of interest that users tend to concern about [15]. Tweets mining can be a challenging task, because the natural characteristics of tweets, such as short-length (140 characters), sparsity, noise, and time sensitivity (topic drift), contribute to complex operations [2][3].

Many researchers agreed that supervised classification, such as TFIDF - Naïve Bayes, was applicable for User Interest Automatic Detection from the tweets of users, the result of which may be desirable and acceptable [2]. However, a technical gap caused by this approach is obvious. Most supervised approaches are reliant on heavy annotation and human transcription - labelling training data. As for dynamic Twitter dataset, that is usually with large volume and velocity, the topic drift makes the work more tedious and time-consuming [13][5]. Moreover, a general opinion is that many supervised models are either required considering strategies to the avoidance of overfitting or underfitting issue, or donating extra time on training models (learning). Thus, this approach is seemingly inefficient and unpractical.

A research problem is how to efficiently detect user's interest from tweets, with the current text mining techniques. Other researchers proposed that the adoption of unsupervised clustering schemes, such as K-Means, DBSCAN and Hierarchical Agglomerative Clustering (HAC), may be an optimal solution to clustering UGC into suitable domains, such as sports or education, based on the preferences of users [4] [5] [19]. However, their current studies were no actual implementation of the clustering techniques, and also there was no explanation on the performance of these clustering techniques filtering on real Twitter's dataset. There is a lack of compelling evidence that the clustering schemes proposed by the current studies are suitable to detect user's interests from tweets efficiently and effectively.

In this research, we aim to propose an experiment to establish a model to enable the three clustering schemes performed in a real scenario. The research question is: how could establish a model for K-Means, DBSCAN, and HAC algorithms to a comparative analysis for exploring their clustering performance and limitations in an unlabelled Twitter's dataset? The new knowledge is discovered during the experimental process, that is involved with three aspects: TF-

IDF converts the tweet corpus into a sparse Document-Term Matrix for similarity calculation, but the similarity computation between two vectors may be dominated by zero values, which makes less distinction for both vectors; the real topic is uninterpretable by the Top-15 features in the best cluster; the pre-processing step lacks the functions to handle informal expressions, indirectly harming the cluster quality. Our research is based on artefact-oriented, thus the tangible outputs include the model and resultant data. The functionalities of this model contain tweets pre-processing, K-Means, DBSCAN, HAC, Silhouette Coefficient evaluation and virtualizations. The resultant data includes: Silhouette Score (Average) for three clustering model; Silhouette Plot for the best model; Top-15 Features and User Group in the best cluster.

The objective of this research will present a comparative study for K-Means, DBSCAN and HAC clustering techniques. This study is underlying the experimental data analysis for identifying their performance and limitations to cluster on a real Twitter's dataset. The expected outcome is that the selected schemes may be highly effective and efficient in clustering tweets. The evaluation criterion to this model is when the Silhouette Score (Average) is more than 0.8, the model is applicable to a real scenario, because it indicates the high-quality cluster generated, which could be beneficial to topic interpretability. As for the cluster generated by the best model, the real topic could be easily identified without ambiguity.

The structure of this research contains 4 sections. The *Related Work* will introduce the process of how the discovery of the technical gap from supervised methods, and methodology of the three clustering approaches. The *Research Method* section will elaborate on how the experiment is designed for achieving the research objective. The *Result and Discussion* section will present the performance and limitations of the three approaches compared by Silhouette Coefficient, the important findings, and other potential factors that can affect the experimental result. The *Conclusion* section will summarize the research findings, model limitations, improvement suggestion and future work.

II. Related Work

In recent years, the large-scale social data of Twitter has been stimulated to many researchers to launch different experiments to perform tweets mining. As tweets (UGC) are open-source that hides the preferences of users [15], the majority of research is launched for business purposes, such as improving personalized recommendation systems to push highly relevant topics to users [24].

A. Supervised Classification

Many researchers are commonly acknowledged that the supervised classification can be effective to automatically detect preferences of users from tweets. Supervised classification is a technique that can 'learn' patterns from the training set and classify unseen data (testing set) [23]. An experiment was conducted to validate various hybrid models (supervised term weight approaches) [2]. This experiment employed a novel Term Weight method, i.e. SW, to combine with SVM, Decision Tree, KNN and logistic regression methods, to perform classification on Sanders dataset, a collection contained 4929 tweets manually classified into four categories. The result proved the hybrid models can provide an exceptional result when performing tweets mining. Research also investigated the performance of using pure and simple supervised algorithms, i.e. Support Vector Machine and Naïve Bayes [8]. The research was performed on sentiment analysis from tweets written in Portuguese during the 2013 FIFA Confederations Cup. The dataset was manually labelled ahead. The result indicated that the performance of both classifiers to detect sentiment polarity were satisfactory for Portuguese tweets.

B. Limitations of Supervised Classification

However, there have many studies showed that supervised classification on the Twitter dataset is inefficient and unpractical. The performance of a supervised classified is associated with the quality of training data. The pre-classified tweets (Training Data) are required to be appropriately labelled before they are donated to a classifier. In order to prepare a high-quality training set, it usually requires experienced data practitioners to complete, which is a tedious and time-consuming process [13] [18]. Also, by the statistic, there are around 6,000 tweets per second on average, which equates to approximately 500 million tweets per day [22]. The supervised classification can only handle the extracted data from a specific time interval [6]. It indicates that this approach cannot be adaptive to topic drift issue, and its analysed results are unable to reflect the most current topics (i.e. user topic of interests). As for applying this method, it is inevitable to deal with overfitting or underfitting issue when adopting the supervised approach. In reality, when a dataset is not labelled, there is no way to perform supervised classification. As a result, using supervised classification on a Twitter dataset is seemingly inefficient and unpractical.

C. Unsupervised Clustering

Some studies have reported that cluster analysis of tweets is highly applicable to this kind of dataset, on account of two reasons: (1) the large volume of data for a high-quality training set is unpractical to manual labelling, but it is not required to clustering schemes; (2) the nature of tweets may have unforeseen groups, but it carries the important nuggets of information, which can be extracted by unsupervised clustering methods [10]. Clustering is the process that groups data instances in a relatively similar cluster, without the need for understanding of data structure and class labels [12]. In a large volume of the text-based dataset, clustering is reliant on proximity measure to find patterns between two documents [9]. A study analysed a range of literature using clustering algorithms, including k-Means, k-Medoids, Hierarchical Agglomerative Clustering (HAC), c-Means, DBSCAN with various distance measures, such as Euclidean Distance Measure, clustering features and evaluation methods (e.g. Silhouette Coefficient) [5]. This literature contributed to the selection of clustering techniques, but this research aimed to discover standardised techniques to evaluate the clustering quality of texts. The research result indicated K-Means, DBSCAN and HAC may be optimal candidates to cluster text-based datasets. This idea was supported by authors [4] [19]. In their research, TF-IDF was widely used for text-based vectorization and Silhouette Coefficient was the primary evaluative method. However, these key studies only proposed a theoretical clustering methodology and did not test on a real Twitter's dataset. In other words, based on the current studies, it is inconclusive that if the k-Means, DBSCAN and HAC approaches would be appropriate to cluster tweets.

D. Research Motivation

A heuristic to the research is stemmed from the advantages of the previous studies which has provided the three possible options for tweets clustering. TF-IDF is a useful algorithm that can convert text-based contents into vectors for proximity measure. Silhouette Coefficient is for model performance evaluation. In this research, we aim to develop a prototype that

will equip with these features, enabling the three clustering schemes on a real Twitters' dataset (unlabelled), thereby exploring their performance and limitations.

III. RESEARCH METHODS

A hypothesis is that K-Means, DBSCAN and HAC can cluster tweets into different domains (topics), that is used for the analysis of the user's interest. This section will answer part of the research question, presenting the process of how to establish the experimental model. The model is developed by Python language, tapping *sklearn* library to implement the three clustering schemes, TF-IDF and Silhouette Coefficient. The virtualization is implemented by the *matplotlib* third-party library. The model will generate various experimental data to analyse and verify this hypothesis. The novelty of our experimental methodology can reflect on the pre-processing stage. We conduct clustering on a real Twitter dataset, that contains the natural characteristics of tweets, thus we have improved upon the method suggested by authors [13], the aim of which is to ensure that this method can be efficiently and effectively remove the noisy contents in tweets.

A. Experiment Design

The experimental design can be summarized in Fig. 1:

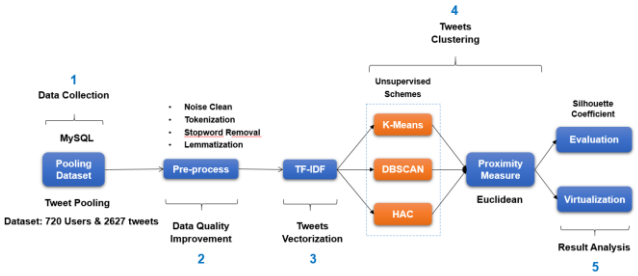

Fig. 1 Experimental Process

B. Dataset

The dataset, given by the AI-Based Data Analysis research group, contains 1,308,469 tweets, created by 965,829 users. The natural characteristics of tweets in the dataset cover a variety of noisy contents, such as non-standard use of English, URLs, user mentions, hashtags, abbreviation, synonym, and misspelling words, and colloquial expressions. The collective time was from 21-03-2006 to 22-12-2012.

C. Approach

1. Data Collection: Pooling Dataset

MySQL technique is used to fetch 720 users and their tweets (2627). Tweet-pooling creates a document that corresponds to a tweet, thus 2627 documents generate.

2. Data Quality Improvement: Pre-processing

This research only focuses on English Tweets, because it constitutes around 50% of all tweets on Twitter's platform [25], so that we will remove non-English words in the pre-processing step. The following pre-processing steps are reliant on the method proposed by [13], the aim of which can improve data quality. We improve some of the steps, based on the experimental dataset characteristics. Before text-based tweets are categorized, we donate them to be pre-processed as the below steps cumulatively:

1) Noise Elimination

- Eliminate non-English tweets that are not in ASCII Decimals 0 - 127
- Remove 'RT' and '&' and user mentions @
- Remove words concatenated with numbers, such as BYG23IM
- Remove emails and URLs (e.g. start with www or http)

2) Tokenization

- Decipher tweets into single words (index)
- Eliminate delimiters (punctuations) and digits
- Lowercase terms

3) Stopwords Removal

- Remove non-meaningful words in a tweet, such as this, which, or in.
- Eliminate when the length of a word <= 2

4) Lemmatization

- Map and decompose various forms of a word to its basic form

3. Tweets Vectorisation: TF-IDF

TF-IDF is the abbreviation of Term Frequency-Inverse Document Frequency, widely used for vectorizing text-based contents in text mining task [4] [11]. TF can capture the importance of a term in the document. IDF can measure the importance of a term that rarely occurs in a corpus. The equation is cited by [11], which can be defined by (1):

$$TF - IDF_{ij} = tf_{ij} \times \log\left(\frac{N}{df_i + 1}\right) \quad (1)$$

tf_{ij} The number of times term i that appears in document j.
N The number of documents in the entire corpus.
df_i The number of documents that contain term i, where the term is at least occurred once.

Tweets are represented as TF-IDF feature vectors, where each vector represents as a set of data points in the n-dimensional space (n indicates the size of the corpus defined by vocabularies in the tweets collection). The weight of a point is calculated by Eq. (1). A Document-Term Matrix consists of vectors [14], which is exampled as Fig. 2.

$$\left\{\begin{bmatrix} & T_1 & T_2 & \dots & T_t \\ D_1 & W_{11} & W_{21} & \dots & W_{t1} \\ D_2 & W_{12} & W_{22} & \dots & W_{t2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ D_n & W_{1n} & W_{2n} & \dots & W_{tn} \end{bmatrix}\right\}$$

Fig. 2 Document-Term Matrix

D_n A single tweet *n* in the collection
T_t A term *t* in the corpus
W_{tn} A weight of term *t* in a single tweet *n*

4. Tweets Clustering

a) K-Means

K-Means clustering is a quantitative vector method, that can partition data points into K clusters, where if points are highly similar (high intra-cluster distance), they are grouped into a cluster [21]. The process can be summarized as below:

1) Select K centroids as the initial clusters arbitrarily.

2) Iterate to cluster points

- Allocate the data points to clusters, where each point is close to a centroid.
- Re-locate the centroids based on the mean of points in the clusters.

3) Converge when each point is optimally clustered

The method is mandatorily required to define the number of clusters (K value). A data point is the vector of a tweet.

b) DBSCAN

DBSCAN is an abbreviation for Density-based Spatial Clustering of Applications with Noise, which is an unsupervised method [14]. It is applied for dividing the heterogeneous tweets into clusters/groups, that contain semantically and syntactically similar contents. If tweets lie alone in high-density regions, they are linked and grouped together; otherwise, tweets may be assigned to in low-density regions, marking as outliers or noisy points. Two data-dependent hyperparameters can affect its performance:

- **Min_Samples**: minimum number of data points to define a cluster.
- **Epsilon (ε):** a range specifies the neighbourhoods, where points in the range (inclusive) are considered to be neighbours.

A vectorized tweet is represented as a data point. In DBSCAN clustering, the three important categories of points will be generated (Fig. 3):

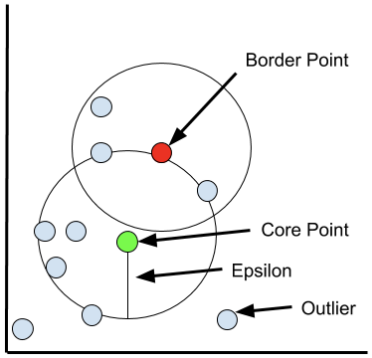

Fig. 3 DBSCAN Clustering Example

- **Core point**: a centre point if the Min_Samples number of points (include itself) in its neighbouring area, within the radius eps. Centre points are connected.
- **Border point**: a border point defined by if it can be reachable from a core point, but the surrounding area contains the number of points that are less than Min_Samples.
- **Outlier**: an outlier defined by if it cannot be reachable from any core points.

c) Hierarchical Agglomerative Clustering (HAC)

The agglomerative is the most common category of hierarchical clustering, which is a 'bottom-up' approach. In Fig. 4, each object starts by treating as a singleton cluster [17]. The nearest pairs of sub-clusters are successively merged as one cluster based on similarity. The iterations will be converged when the ultimate cluster merged by all sub-clusters. The structure can be represented by a tree-based graph - Dendrogram. HAC is a non-parametric algorithm. A single tweet (vector) can be identified as an object.

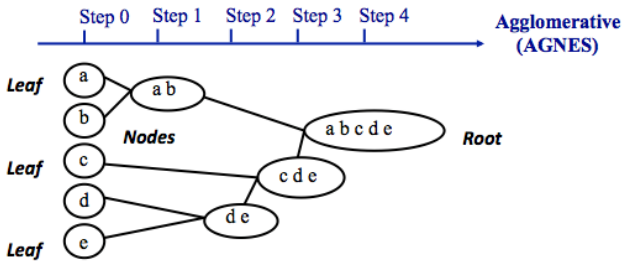

Fig. 4 Agglomerative Clustering Example

d) Proximity Measure

Euclidean Distance Metric is applied to the three clustering schemes to determine the similarity between two data nodes (vectors). The formula is referred to [24], which can be defined by (2):

$$D_{Euclidean}(X,Y) = \sum_{k=1}^{d} \sqrt{|x_{ik} - y_{ik}|^2} \quad (2)$$

X, Y	A pair of vectors (tweets with TF-IDF feature)
d	The size of vector dimension.
$\mathbf{x_{ik}}$, $\mathbf{y_{ik}}$	Weights in vectors of X and Y

5. *Result Analysis*

a) Evaluation

We apply Silhouette Coefficient to evaluate model performance, which is a cluster validity measure [1]. The measurable value of this method ranges -1 to +1. If the evaluative coefficient is close to 1, it implies the object is highly correlated to its belonging cluster, while if the coefficient is close to -1, it indicates that the sample is much closer to the neighbouring cluster, instead of the assigned cluster. The formula is summarized by [1], which can be defined as follow:

$$a(i) = \frac{1}{|C_i|} \sum_{j \in C_i, i \neq j} d(i,j) \quad (3)$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j) \quad (4)$$

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (5)$$

a(i)	The average distance calculated by the data point i to other points in the same cluster. Large a(i) indicates this sample is dissimilar to its cluster.
b(i)	The average distance calculated by the data point i to other points in the nearest cluster. Large b(i) indicates this Sample is dissimilar to its neighbouring cluster.
s(i)	Silhouette coefficient with range [-1, 1]

- *Silhouette Score (Average)*

The evaluative result is generated by this evaluative method, which is a significant measure that can define the performance of a clustering model. The score averages all clusters grouped by a clustering model. The experiment aims to present the average score for each clustering model, for performance comparison. The range of this score is in [-1 to 1]. A clustering model can contribute an average score. If the score is more than 1, the best model is defined by the highest silhouette score. If the

score is less than 0 and close to -1, we identify the clustering model may not suitable to detect User Interest based on tweets.

b) Virtualization

In this experiment, we will create 3 types of graphs for the auxiliary analysis to model performance, which can be summarized as below:

- *Silhouette Plot*

This measure aims to explore the silhouette score for each cluster grouped by the best model. It is assistive to identify the best cluster, thereby exploring the Top-15 Features – it could implicitly reflect a topic that users may be concerned about.

- *Top-15 Features*

This measure can assist to detect the Top-15 frequent words in the best cluster generated by the best models. We observe the Top-15 Features for getting insight into the real topic to identify User Interest.

- *User Group*

This measure aims to detect what users are in same interestingness from the best cluster. We can identify whether a clustering model could succeed in group users who hold similar interests.

D. Hyperparameter Detection

Table 1. Experimental Parameter Detection

Schemes	Params	Method
K-Means	K (the number of clusters)	Elbow Method
DBSCAN	Epsilon & Min_Samples	A Grid Matrix
HAC	K	Elbow Method

Before performing tweets mining, it requires to be well-configured the three clustering schemes. As Table 1 shows, the Elbow Method is a heuristic approach, widely applying for the detection of the optimal number of clusters to a text-based dataset. In our research, we detect the optimal value by the balance between Silhouette Score (Average) and Time Expense. A Grid Matrix will create various DBSCAN models with a bag of combinative hyperparameters, that will be evaluated by Silhouette Score (Average). The best combinative hyperparameter will be defined by the highest evaluative score.

IV. Results and Discussion

In this section, we will present the experimental results and findings to answer the part of the research question, that is involved with providing a comparative analysis to explore the performance and limitations of K-Means, DBSCAN and HAC. Also, we will provide solutions to new knowledge generated by our findings.

A. TF-IDF Representation

Table 2. The Size of Document-Term Matrix (DTM)

Original	Improved
n_samples: 2627 n_features: 4578	n_samples: 2627 n_features: 307

As Table 2 shows, the corpus contains 4578 unique vocabularies analysed by TF-IDF, which is not beneficial to efficiency in terms of tweets clustering. We empirically set the upper bound to the features <= 800, where the Top-800 terms ranked by the Term Frequency in decreasing order will be selected for the experiment. After that, the minimum Document Frequency (DF) we set a lower bound to 10, which indicates the ignorance of a term with DF < 10. Finally, we have the Document-Term Matrix (DTM) with 307 features, which is significantly less than the original one. As Fig. 5 shows, an example of two tweets is transformed into DTM through pre-processing and TF-IDF.

Original Tweets

It's so **nice** out. It **almost** makes me want to go for a walk.

How to take a photograph on a 5x4 camera #gear #camera #**howto** https://t.co/nEAC6lam

↓

Processed Tweets

nice almost make want walk

take photograph camera gear camera **howto**

↓

Document-Term Matrix

Index	nice	almost	…	howto
0	0.43	0.43	…	0
1	0	0	…	0.43

Fig. 5 Example of Tweets Transformation

B. Experimental Parameter Detection

The experiment was initially required to define the optimal hyperparameters to K-Means and HAC, and we utilized the Elbow method to explore. In Fig. 6, We evaluated both algorithms with K ranged in [2, 25], where time expense in seconds (efficiency) and Average Silhouette Score (effectiveness) were considerations to determine the optimal K value. We expected to this value that can enable the model contributed to relatively higher Silhouette Score (Average), but it was with a small number of clusters (<= 10) and less time cost. As for DBSCAN, we donated Grid Matrix to create various models with different combinations of Epsilon and Min_Samples. The optimal combination was decided by the highest Silhouette Score (Average).

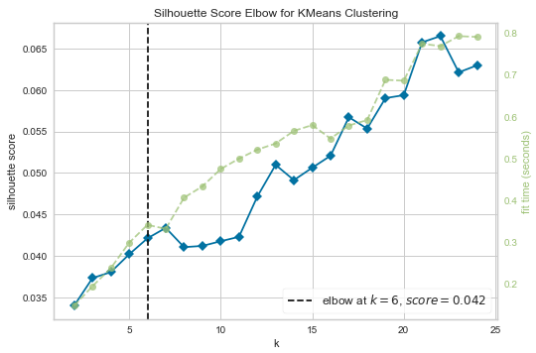

Fig. 6 Optimal K to K-Means

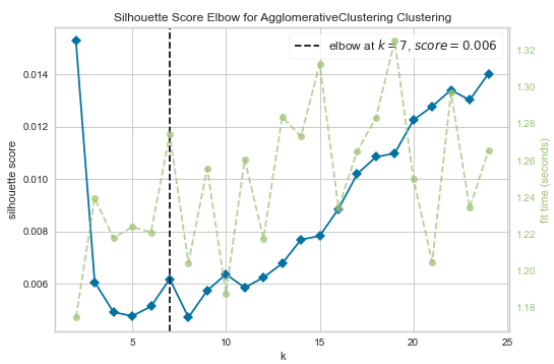

Fig. 7 Optimal K to HAC

As Fig. 6 and Fig. 7 show, the Elbow method automatically detected the optimal values to K-Means and HAC, with 6 and 7 respectively. We have observed that the increase of K value is associated with the growth of Silhouette Score. It is a trade-off decision, which means if we expected to create the high intra-distance cluster, the number of clusters should be given as much, but it harms efficiency.

Table 3. DBSCAN Grid Matrix

Grid Matrix		Min_Samples				
		2	3	4	5	
Epsilon	0.1	-0.0291	-0.1036	-0.1375	-0.154	Average Silhouette Score
	0.2	-0.0285	-0.1023	-0.1375	-0.154	
	0.3	-0.0261	-0.101	-0.1345	-0.1535	
	0.4	-0.018	-0.0964	-0.1301	-0.1526	
	0.5	-0.0077	-0.0807	-0.116	-0.1403	
	0.6	0.0077	-0.064	-0.1001	-0.1298	
	0.7	0.0361	-0.0278	-0.0625	-0.0965	
	0.8	-0.02	-0.015	-0.0115	-0.0158	
	0.9	-0.0602	-0.0197	-0.0103	-0.0149	
	1.0	0.0177	0.0235	0.0235	0.0235	

As Table 3 shows, the combinative hyperparameters [Epsilon = 0.7 & Min_Samples = 2] enable DBSCAN to generate the highest number of Average Silhouette Score.

C. Model Comparison by Silhouette Score (Average) [SC (Avg)]

Epsilon = 0.7 & Min_Sample = 2
1585 Noise Points

Table 4. SC (Avg) for the three optimal K

	K-Means	DBSCAN	HAC
Optimal K	6	233	7
SC (Avg)	0.0421	0.0361	0.0062

Table 5. SC (Avg) for K = 6

K = 6		
	K-Means	HAC
SC (Avg)	0.0421	0.0052

Table 6. SC (Avg) for K = 7

K = 7		
	K-Means	HAC
SC (Avg)	0.0434	0.0062

Table 7. SC (Avg) for K = 233

K = 233		
	K-Means	HAC
SC (Avg)	0.1421	0.1381

As Table 4 shows, K-Means is the best model compared with DBSCAN and HAC. DBSCAN is ranked the second, but it contributes to 1585 noise points and the largest number of clusters (233). HAC is the worst model evaluated by a small number of clusters (7). As Table 5 - 7 show, in terms of K-Means and HAC, the increase to K is associated with the growth of SC (Avg), which means both models are significantly benefited from a large number of clusters. This may indicate the sparse DTM is preferable to form by many sub-clusters. DBSCAN cannot directly configure. K value, but it is defined by the Epsilon = 0.7 and Min_Samples = 2. Hence, we only compare K-Means and HAC to evaluate their performance with different optimal K values.

D. Quality Cluster Detection

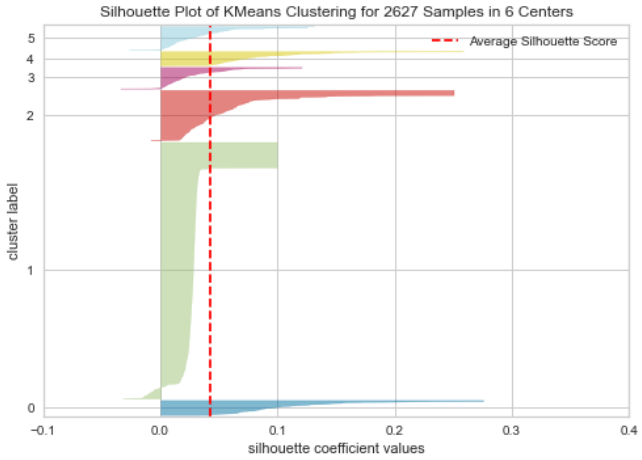

Fig. 8 Silhouette Plot

Fig. 8 is assistive to explore the performance of K-Means (the best model) with the optimal K = 6, where the red line is the SC (Avg). The Silhouette Plot is plotted for exploring each cluster grouped by this model. The useful information can be summarized:

- The sample distribution is unaveraged to the 6 clusters.
- Cluster 1 is the larger group, but numerous samples are less than SC (Avg)
- Cluster 0 and 4 are the better clusters, as most samples are over than SC (avg)

The importance of cluster analysis is associated with topic interpretability [7]; thus, we select the best clusters 0 and 4 for virtualization.

E. Quality Cluster Analysis

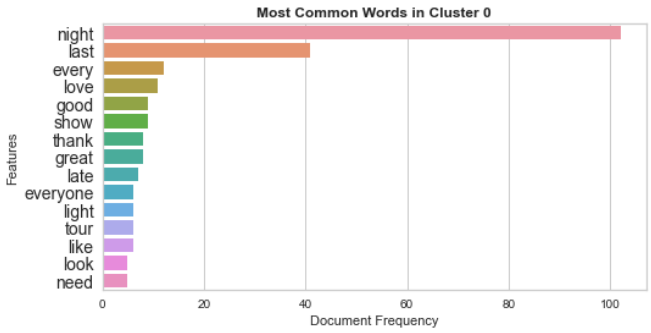

Fig. 9 Top-15 features for cluster 0

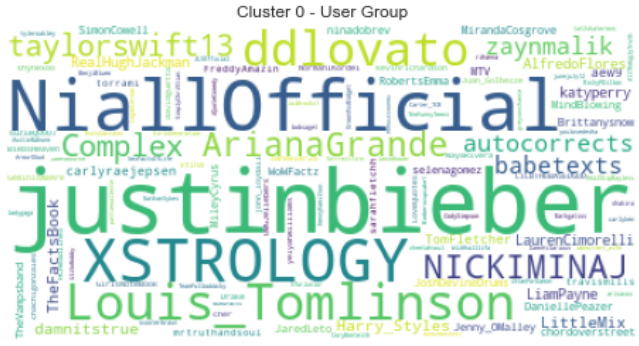

Fig. 10 Users in cluster 0 (wordcloud)

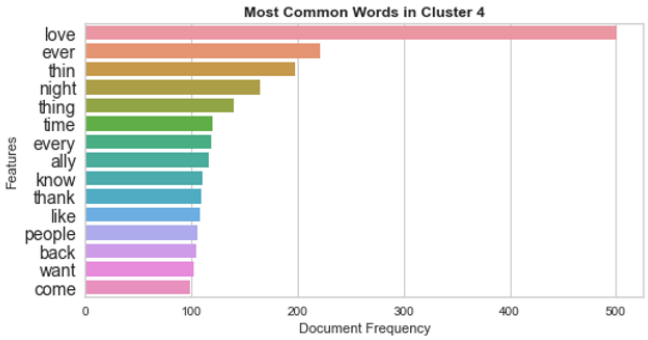

Fig. 11 Top-15 features for cluster 4

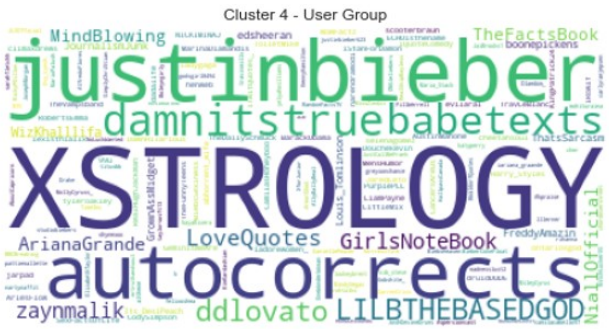

Fig. 12 Users in cluster 4 (wordcloud)

As Fig. 9 and Fig. 11 show, it is difficult to interpret the actual topic from the Top-15 features, as the most features are non-meaningful, such as 'good', 'show' or 'thank'. However, as for some meaningful words, it is difficult to discover the relationship between these words. For example, as for terms 'tour' and 'night' in cluster 0, we could only assume that this topic is related to Night Tours or Night Activities, which contributes to ambiguity in topic interpretability or which cannot be accurately interpreted as a real topic.

As Fig. 10 and Fig. 12 show, we can easily identify users who are interested in the two topics, and also a user can be interested in many topics. For example, the users, named 'XSTROLOGY' and 'justinbieber', appears in both clusters.

F. Result Analysis

This section will answer the research question based on the data analysis, and comment on the performance and limitations of the three schemes.

1. *Model Performance*

As Table 4 - 7 show, the three models are not applicable filtering clustering currently, as their contributions in Silhouette Score (Average) is lower than our expectation (> 0.8 is appropriate), which is rejected to the initial hypothesis. Despite K-Means is better than DBSCAN and HAC, it is still required to improve in further research.

2. *Model Limitations*

The three models are sensitive to points density. However, we have proved that the sparse DTM contributes to many points with low-density. As Table 2 shows, the size of the dimension to a vector is 307, but a processed tweet only contains a few words. It indicates that the distribution to points is sparse, which may significantly harm the performance of clustering models. The experiment proves that TF-IDF is not suitable for vectorizing tweets, as it will contribute to the sparse matrix.

3. *Importance Findings (New Knowledge)*

TF-IDF: The failure of TF-IDF to weight term is caused by the natural characteristics of tweets, such as short-length and sparsity. TF-IDF is sensitive to document length, but in such as short-text microblogging message, the term frequency to a term may be 1 or 2, which may adversely affect the weight of a term. Also, regardless of vectors or DTM, the sparsity issue is inevitable. This is an unexpected result. The solution can replace TF-IDF to doc2Vec, the method of which is capable of fixing the length of the vector and enabling it to be dense [16].

Pre-processing: In the current model, there is no strategy to handle informal expressions, such as elongated words (e.g. reaaaaally ⇔ really). It indicates that there has unnecessary information in the tweet, which will contribute a negative impact on cluster quality. The solution can tap Bing Spell Check API, which is an algorithm that can automatically detect the misspelling term, and providing suggestion to correct this word [13].

Topic Interpretability: the best model (K-Means) is unexpected to generate low-quality cluster, as it contributes to the low SC(Avg), which is rejected to our initial expectation. As Fig. 9 and Fig. 11 show, the Top-15 Features in its best cluster are not useful to recognize the real topic and difficult to deduce the user's interests. The solution can replace top-features to top-hashtags based on their frequency [15].

4. *Influence by Other Factors*

The performance of K-Means and HAC can vary considerably, which will be caused by the difference of hyperparameter configuration. As Fig. 6 and Fig. 7 show, the increase of K value is associated with the growth of SC (Avg). It seems a trade-off decision. If we only consider forming many high intra-distance clusters, it is encouraged to set a large K value, but it harms efficiency. This is an extra factor that can affect our experimental result. This effect will be donated to further research for exploration.

5. *Experimental Comparison*

Our experimental results are inconsistent with the results proposed by [4] [19]. The reason is caused by the different datasets. Their datasets were News20, Reuters, Emails, and UCI KDD Archive. We apply the three model to real Twitter's dataset. The weakness of TF-IDF cannot be assistive to the three models to generate a desirable result, as the real tweets are short-length and sparse.

V. CONCLUSION

In this experiment, K-Means, DBSCAN and HAC clustering approaches are unable to clearly identify meaningful features for topic interpretability and their Silhouette Scores (Average) are less than our expectation (< 0.8). Despite K-Means is the best model compared with DBSCAN and HAC, it needs to improve significantly. Three important research findings are beneficial to advance the current state of knowledge. The first finding is involved with tweets vectorization. As the natural characteristics of tweets, such as short-length and noise, TF-IDF contributes to a sparse Document-Term Matrix with tweet's vectors. However, the three clustering schemes are sensitive to the density of points. Low-density matrix substantially harms model performance in clustering tweets. Hence, the improvement of tweets vectorization may be a pathway that enhances the clustering result. Another research finding is related to the improvement of the pre-processing step. There are no solutions to handle informal expressions and misspending, such as the elongated words (reaaaaally ⇔ really), in the current design.

The noisy contents are not helpful to improve model performance. Thus, it requires to take measures to enrich the functionalities of the pre-processing step, enabling it to deal with the diversity of tweets and ensuring the data quality. The third finding is involved with topic interpretability, the experimental result of which indicates the difficulty of interpreting the real topic based on the Top-15 Features, because the low-quality cluster contributes to many non-meaningful words, which is not beneficial to topic translation.

The novelty of the research is to valid unsupervised clustering methods to filter tweets from an unlabelled dataset. Compared with supervised classification, the method is more efficient, as there is no need for tweets annotation (label training data). Although the research result indicates the three models are less effective to clustering tweets, we have discovered the way to improve, which may imply unsupervised approaches are applicable to deduce user's interests, in tweets clustering, by an iterative process to improvement. The implication of this research is to open a novel way to efficiently cluster tweets. Our research method presents the entire process, which could be the reference to other data practitioners who aim to perform text-mining on large-scale micro-blogging messages. They could improve on the experimental approach and in-depth exploration of this orientation.

The suggestion to the improve clustering performance is based on our important findings. As TF-IDF fails to assist clustering schemes to filter tweets, we suggest adopting doc2Vec to replace TF-IDF for tweets vectorization. Doc2vec can fix the length of the vector and convert the sparse vector to be dense. The enhancement to pre-processing step can be embedded Bing Spell Check API for improving data quality, where this method can automatically correct the word. Topic Interpretability can consider replacing top-features to top-hashtags.

ACKNOWLEDGMENT

This research idea is based on Dr. Xu Yue and Dakshi Kapugama Geeganage, where the process is encouraged and supervised by them. The author is thankful to their valuable comments and constructive suggestion.

REFERENCES

[1] Aranganayagi, S., & Thangavel, K. (2007). Clustering categorical data using silhouette coefficient as a relocating measure. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)* (Vol. 2, pp. 13-17). IEEE.

[2] Alsmadi, I., & Hoon, G. K. (2019). Term weighting scheme for short-text classification: Twitter corpuses. *Neural Computing and Applications, 31*(8), 3819-3831.

[3] Alvarez-Melis, D., & Saveski, M. (2016). Topic modeling in twitter: Aggregating tweets by conversations. In *Tenth international AAAI conference on web and social media*.

[4] Bafna, P., Pramod, D., & Vaidya, A. (2016). Document clustering: TF-IDF approach. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (pp. 61-66). IEEE.

[5] Crockett, K. A., Mclean, D., Latham, A., & Alnajran, N. (2017). Cluster Analysis of twitter data: a review of algorithms. In *Proceedings of the 9th International Conference on Agents and Artificial Intelligence* (Vol. 2, pp. 239-249). Science and Technology Publications (SCITEPRESS)/Springer Books.

[6] Curiskis, S., Drake, B., Osborn, T., & Kennedy, P. (2020). An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit. *Information Processing and Management*, *57*(2). https://doi.org/10.1016/j.ipm.2019.04.002

[7] D'Andrea, E., Ducange, P., Bechini, A., Renda, A., & Marcelloni, F. (2019). Monitoring the public opinion about the vaccination topic from tweets analysis. *Expert Systems with Applications, 116*, 209-226.

[8] Firmino Alves, A. L., Baptista, C. D. S., Firmino, A. A., Oliveira, M. G. D., & Paiva, A. C. D. (2014). A Comparison of SVM versus naive-bayes techniques for sentiment analysis in tweets: a case study with the 2013 FIFA confederations cup. In *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web* (pp. 123-130).

[9] Godfrey, D., Johns, C., Meyer, C., Race, S., & Sadek, C. (2014). A case study in text mining: Interpreting twitter data from world cup tweets. *arXiv preprint arXiv:1408.5427*.

[10] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, *1*(12), 2009.

[11] Ghosh, S., & Desarkar, M. S. (2018). Class specific tf-idf boosting for short-text classification. *Proc. of SMERP*, *2018*.

[12] Han, J., & Kamber, M. (2012). Data Mining Concepts and Techniques, ed Elsevier Inc.

[13] Ibtihel, B. L., Lobna, H., & Maher, B. J. (2018). A semantic approach for tweet categorization. *Procedia Computer Science*, *126*, 335-344.

[14] Indah, R. N. G., Novita, R., Kharisma, O. B., Vebrianto, R., Sanjaya, S., Andriani, T., ... & Rahim, R. (2019). DBSCAN algorithm: twitter text clustering of trend topic pilkada pekanbaru. *In Journal of Physics: Conference Series* (Vol. 1363, No. 1, p. 012001). IOP Publishing.

[15] Jipmo, C., Quercini, G., & Bennacer, N. (2017). FRISK: A multilingual approach to find twitter interests via wikipedia. Lecture Notes in Computer Science (including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10604, 243–256. https://doi.org/10.1007/978-3-319-69179-4_17

[16] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning* (pp. 1188-1196).

[17] Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*.

[18] Nazura, J., & Muralidhara, B. L. (2017). Semantic classification of tweets: A contextual knowledge based approach for tweet classification. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-6). IEEE.

[19] Nithya, P., Umamaheswari, R., & Shanthi Dr, N. (2014). An Enhanced Similarity Computation for Document Clustering Approaches. *International Journal of Science and Engineering Research (IJSER)*, *2*(10).

[20] Reddy, T. R., Vardhan, B. V., & Reddy, P. V. (2016). Profile specific document weighted approach using a new term weighting measure for author profiling. *International Journal of Intelligent Engineering and Systems, 9*(4), 136-146.

[21] Ravindran, R. M., & Thanamani, A. S. (2015). K-means document clustering using vector space model. *Bonfring International Journal of Data Mining*, *5*(2), 10-14.

[22] Ruz, G. A., Henríquez, P. A., & Mascareño, A. (2020). Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers. *Future Generation Computer Systems, 106*, 92-104, evolving terms and slangs (informal expressions)

[23] Stephens, D., & Diesing, M. (2014). A comparison of supervised classification methods for the prediction of substrate type using multibeam acoustic and legacy grain-size data. *PloS one*, *9*(4), e93950.

[24] Seo, Y. D., Kim, Y. G., Lee, E., & Baik, D. K. (2017). Personalized recommender system based on friendship strength in social network services. *Expert Systems with Applications*, *69*, 135-148.

[25] Wilson, T., Stanek, S. A., Spiro, E. S., & Starbird, K. (2017). Language Limitations in Rumor Research? Comparing French and English Tweets Sent During the 2015 Paris Attacks. In *ISCRAM*