Tweet Filtering Based on User Interests Using Doc2Vec and Unsupervised Clustering

Xinruo Wang
chobitssinsin@yahoo.co.jp
Science and Engineering Faculty
Queensland University of Technology
2 George Street, Brisbane, 4001, QLD, Australia

***Abstract*—Microblogging sites have been essential sources of information for users. Twitter is especially popular with 500 million tweets generated daily. However, due to the increasing volume of tweets available and the fact that those tweets are unannotated, it is difficult for users to find relevant tweets. Therefore, it is necessary to use data mining techniques to automatically detect user interests from their tweets and find relevant posts for them. This research aims to investigate how to filer tweets based on user interests using doc2vec, TF-IDF feature representation methods and unsupervised clustering techniques. Many researchers have applied the traditional TF-IDF feature representation method and supervised clustering techniques to group tweets with similar topics together, but they usually result in poor classification accuracy because tweets are short and noisy. Moreover, supervised clustering requires labelled data, which are unavailable in real situations. To bridge the gap, this research implements doc2vec, which can be a better feature representation method, and compares the performance with TF-IDF using two unsupervised clustering techniques, namely k-means and hierarchical agglomerative clustering. The results show that doc2vec outperformed TF-IDF regarding clustering quality using intrinsic evaluation measures. In particular, doc2vec combined with k-means delivered the best performance. We also demonstrated a method to interpret user interests in each cluster produced by the best model.**

Keywords—document clustering, do2vec, TF-IDF, unsupervised clustering, microblogging

I. Introduction

Microblogging services have been playing an important role in online communication in recent years [9]. Twitter is a particularly popular microblogging site where 350 million daily active users share their thoughts, news and feelings though short texts called tweets. Due to a large volume of tweets generated daily, it is considered as a rich collection of textual data, which could potentially reveal valuable insights about users. In fact, many researchers suggest that tweets are a useful indicator of user preferences as people tend to post what they are interested in [10]. If user interests can be derived from tweets, it is beneficial not only for businesses, but also for users because they can find people with similar interests. However, most of the existing methods rely on hashtags as keywords to find relevant tweets. These methods are inefficient due to information overload and unannotated data [5]. To address these issues, various studies have dealt with automatic tweet categorisation.

A common method to find user interests from text is to use Bag of Words (BoW) and a term weighting scheme. BoW treats a document as a bag of words and term weighting applies statistical regularities to allocate a weight to each word [15]. Among those, the term-frequency inverse-document-frequency (TF-IDF) approach is employed in many studies for keyword extraction in microblogs [4][14]. As the name indicates, the underlying assumption is that a word which more frequently appears in a document and less frequently appears in the corpus is considered as a keyword. [14]. However, these traditional approaches have two major problems. They ignore relationships between terms such as ordering and synonyms, as well as semantics of the terms [13]. Moreover, when it comes to tweets, they are often short and noisy resulting in poor classification accuracy [9]. To overcome these weaknesses, [13] introduced an alternative method called doc2vec (Paragraph Vector) to represent documents as dense vectors. When this method was applied to detect similar paragraphs in the same query from a search engine, it obtained a much higher accuracy than BoW [13]. Moreover, this method is applicable to variety of texts from a sentence to a document. Therefore, doc2vec has a potential to be a better choice of user interest extraction for tweets. Once tweets are vectorised, clustering techniques can be applied to group tweets with similar interests together.

Following the appearance of this new approach, [5] combined doc2vec with different clustering techniques to evaluate model performance of tweet data. However, the limitation of the research is that it uses supervised learning, which requires ground truth labels for evaluation.

In order to deal with the problem, this research aims to investigate how to filer tweets based on user interests using doc2vec, TF-IDF and unsupervised clustering techniques. This is important because tweets are unannotated in nature; thus, unsupervised learning is more in line with real scenarios. This research will answer questions such as, if doc2vec can capture user interests better than TF-IDF, what is the most effective clustering method to group tweets with similar interests together? The new knowledge generated from this study is a comprehensive model to extract user interests from tweets and cluster similar tweets together. This is an artefact-oriented research with the final outputs to be a prototype system which incorporates the best clustering model and filters relevant tweets for users.

The objective of this project is to demonstrate the effectiveness of the doc2vec method and identify the best clustering technique to cluster tweets together. The expected outcome is a performance matrix with combinations of different feature representation methods and clustering techniques.

Out contribution can be summarised as follows:

1. Propose a tweet clustering model using doc2vec and unsupervised clustering techniques, which do not require topic annotation
2. Propose a method to interpret cluster topics using the doc2vec and k-means clustering model

The rest of the paper is organized in the following orders; Section 2 for literature review, Section 3 for research methods, Section 4 for results and discussion and Section 5 for conclusion.

II. RELATED WORKS

We organise related work on tweet clustering into two areas. Firstly, we explain traditional document clustering approaches, then we describe recent advances in feature representation methods and how they can be applied to tweet clustering.

A. Document clustering and topic modelling

To apply document clustering to tweets, the first step is to convert the texts into collections of vectors which represent word occurrence. The vector values are assigned using one of many feature representation methods including the most common method, TF-IDF. As the size of the vector space is determined by the number of unique words in a corpus, there is often a cut-off value to only keep words with high values for use [17]. After feature vector matrices are obtained, clustering algorithms such as k-means and hierarchical clustering can be applied to cluster tweets with similar interests.

For example, [8] developed a tweet clustering model using TF-IDF and two clustering methods k-means and Non-negative Matrix Factorisation (NMF), and examined the performance using approximately 30,000 tweets related to the World Cup in 2014. The results revealed that both clustering techniques had similar results. The novelty of this research is that it proposed a new model to overcome a shortcoming of tweet clustering, that is, irrelevant tweets being clustered together due to noise. In detail, the model introduced a noise removal algorithm to eliminate outlier tweets from the corpus before applying clustering. This algorithm utilises DBSCAN clustering technique with an assumption that noisy tweets are those which are not close to any clusters.

Another study which used TF-IDF and clustering techniques is [12], where various clustering techniques such as k-means, k-centroid, DBSCAN and NMF were compared to see which outperforms for tweet topic categorisation related to a recent earthquake in Nepal. The achievement of the study is the evaluation method, which used intrinsic measures such as within-cluster variance and between-cluster variance to compare how well clusters are separated from each other. The result indicated that all of four clustering methods had similar performance, but NMF was superior in terms of simplicity in topic interpretation. However, the topic interpretation was compared by manual inspection of frequent terms in each cluster.

Topic modelling such as Latent Dirichlet Allocation (LDA) and Author-Topic model (ATM) is also a method to detect topics from documents. It is capable of finding patterns of words and represents topics as a distribution of words. Each document is given probability distribution of various topics. Topic modelling has also been an active research topic in topic categorisation of microblogging data.

For instance, [3] proposed a new method of tweet pooling, where tweets are aggregate in conversational level before being trained using topic models. The approach aims to deal with a problem of traditional topic modelling techniques, which is sensitive to noise and document size. While noise can be removed by text pre-processing steps, the problem of length remains. Therefore, it is necessary to supply the deficit by concatenating tweets in a way that they share similar topics. When the new pooling approach was examined using LDA and ATM, it outperformed existing pooling techniques regarding clustering quality. The achievement of the study is that it is able to produce better clusters more quickly especially using ATM.

B. Neural network embedded models

A large number of topic clustering on microblogging sites adopt TF-IDF to represent text as a matrix of vectors. However, the limitation of this method is that it ignores the ordering and semantics of words, and produced matrices are high-dimensional and sparse. To solve these problems, [13] proposed a new feature representation method called doc2vec. In this method, documents are converted to dense vectors, which can be trained using a neural network to enable word prediction in documents. It can also map similar words if vector positions of two words are similar to each other. To examine doc2vec is more competitive than BoW models, an information retrieval task was conducted using paragraphs obtained from the top 10 results of 1 million most popular queries on a search engine. Firstly, for each query, triplets of paragraphs were created with two derived from the same query and one from an unrelated query. The task was to predict which paragraphs were from the same query. To measure the performance, distances of paragraphs were computed after applying some feature representation methods including TF-IDF and doc2vec. It was considered as successful, if it computed the distance of first two paragraphs to be smaller than the third one. Finally, the number of successful counts was recorded to compare the performance between the different methods. The result showed that doc2vec performed favourably against BoW in terms of classification accuracy. The advantage of this new method is not only it can learn semantics, but it can also be applied to different length of texts from a sentence to a document. Therefore, doc2vec can be a better solution for tweet topic clustering.

To provide a guidance on the best approach to perform document clustering and topic modelling on microblogs, [5] combined four feature representation techniques including TF-IDF and doc2vec, and four clustering techniques to evaluate their performance on Twitter and Reddit datasets. It also included a LDA topic model for comparison. The study implemented extrinsic performance measures such as Normalized Mutual Information, which require ground truth labels. The results showed a trend that the combination of neural word embeddings and clustering techniques have a competitive advantage over others such as TF-IDF and topic modelling. Specifically, doc2vec combined with k-means works effectively in both Twitter and Reddit datasets. However, the limitation of the approach is that it requires annotated tweets, but raw tweets do not contain topic labels and it usually requires manual work to prepare an annotated dataset.

This study intends to fill the gap by adopting the silhouette coefficient, which does not require labelled data, as the measurement of clustering performance because it aligns more with real scenarios. In addition, it will examine the hypothesis that doc2vec can bring better performance than TF-IDF by making comparisons of the two methods combined with two clustering techniques, namely k-means and hierarchical agglomerative clustering.

III. METHODS

In this section, we describe research steps outlined in Fig. 1, including data collection, feature representations, clustering, evaluation and model implementation.

This research adopts artefact-oriented research method involving model development and performance evaluation. Firstly, tweets were converted to vectors using the TF-IDF and doc2vec feature representation techniques. Secondly, documents were clustered using k-means and hierarchical agglomerative clustering. After four models were constructed, performance validation was conducted using the silhouette coefficient to see how distinct the produced clusters were. With the best model determined, we developed a prototype system to display relevant tweets for users from the same clusters.

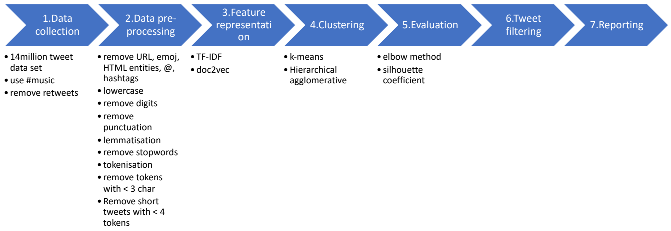

Fig. 1. Research steps

A. Data collection

We used 14million Tweets as the original database, which contains tweets collected via Twitter API from May to June in 2013 [20]. The data was extracted using trendy hashtags found in Hashtag.org as search keywords. The collected tweets often contained popular hashtags used by spams such as #followback, #gameinsight because the research aimed to identify spammy tweets. However, the objective of this project is to filter relevant tweets for users. Hence, we used a highly reliable hashtag #music to extract tweets from the 14million Tweets database. The hashtag was also listed in the top-20 for most popular hashtags in [20]. In addition to this, it also included other tweets posted by the same users because a user may have more than one interest. After tweets were collected, retweets were discarded as they do not provide useful information [1][16]. In total, the number of remaining tweets were 5732, posted by 2419 users.

B. Data preprocessing

After data collection, the next step is to perform natural language pre-processing. The purpose is to clean the data, reduce dimensions of words and only keep important words which have potential to represent topics. Firstly, the URL, reserved words (FAV), emojis, user mentions and hashtags were removed. Secondly, tweets were lowercased, and digits were discarded. Thirdly, we removed stop-words and punctuations, and performed lemmatisation. In the original work of doc2vec [13], it does not mention if lemmatisation should be applied; however, we applied the technique because it allows the reduction of the number of words which have the same meaning. In the last step, we tokenised tweets into collections of words and removed tokens with less than three characters as they do not contain important meanings. Python libraries including Natural Language Toolkit and genism were used here. After this process, the number of tweets in the corpus was reduced to 3974, posted by 1799 users. The average number of tokens in each tweet was 7.11. The statistics are summarised in Table 1.

TABLE I. STATISTICAL SUMMARY ABOUT THE CORPUS

Number of tweets	3974
Number of users	1799
Average Number of tokens in each document	7.11

C. Feature representation

On the premise of our model, a tweet is represented as a single document and a user is represented as a collection of documents. Therefore, each tweet was converted into a fixed-length vector before applying clustering. We utilised the TF-IDF and doc2vec techniques for feature representation using python library sklearn and gensim.

1) TF-IDF

TF-IDF is defined as (1), where TF denotes number of times a word *i* appears in a tweet *j*, and DF denotes a number of tweets containing *i*. *N* is the total number of tweets in the corpus [19].

$$TF - IDF_{i,j} = TF_{i,j} \times \log\left(\frac{N}{DF_i}\right) \qquad (1)$$

We computed TF-IDF for each tweet against all other tweets. In addition, we removed words that are too infrequent with a document frequency of less than 20 from TF-IDF calculation. 223 features remained after this process.

2) Doc2vec

Doc2vec employs a neural network for training to produce a dense vector for each document. Therefore, there are multiple parameters to be set. This research referred to [5] and set the underlying model to be distributed bag of words, the number of dimensions as 100, content window as five and minimum word count as one. The optimal number of epochs is a key parameter and was explored to see which provides the best evaluation result. This is because the optimal value seems to be dependent on different tweet datasets [5]. The result showed 15 epochs gave the best performance using k-means clustering; hence, this was utilised in the other clustering technique as well.

D. Clustering

There are two clustering methods selected to cluster tweets together including k-means and agglomerative hierarchical clustering. Both of them can be performed with python library sklearn.

1) K-means clustering

The first method is k-means clustering which is the most commonly used approach in data mining field [8]. In this method, *n* instances are clustered into *k* clusters by assigning each instance to a closest centroid. In this study, we used the Euclidian metric to measure the distances between instances and centroids following [5]. Additionally, maximum iterations will be set to 100 for convergence. As for the determination of optimal k, the elbow method was used, which plots the within-cluster sum of squares against consecutive k to see where the value starts to diminish. Equation (2) explains the within-cluster sum of squares, where *k* and *n* denote the number of clusters and the number of datapoints respectively. It calculates the sum of the squared distance between datapoint *x* and its cluster centre *c* for all instances. [12]

$$SSW = \sum_{j=1}^{k}\sum_{i=1}^{n} d(x_i^{(j)}, c_j)^2 \qquad (2)$$

2) *Hierarchical agglomerative clustering*

Hierarchical agglomerative clustering takes bottom-up manner to cluster instances [2]. It has seen applications in many studies [5] [7]. There are two important parameters to consider when applying hierarchical agglomerative clustering, which are similarity measurements between datapoints and between clusters. For these we will use the Euclidian metric to measure the distance of data points and Ward linkage to measure the distance of clusters.

E. Evaluation

In order to determine which feature representation method and clustering algorithm are best to categorise user tweets, it is necessary to have an evaluation measure. As our dataset does not contain class labels, we used the silhouette index to validate our methods. The silhouette index measures the differences of two datapoints within and between clusters [21]. In (3), for a data point *i* in cluster *C*, it calculates the mean intra-cluster distance.

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

On the other hand, (4) calculates the mean distance of *i* to all other points in the nearest cluster.

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

Finally, the silhouette score *s(i)* can be derived from (5).

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

The values for silhouette scores range from -1 to 1, and 1 indicates clusters are well separated. This method has been widely used in clustering validation. For example, [6] applied the silhouette score to measure clustering performance of tweets for k-means and hierarchical clustering.

F. Tweet filtering

Tweet filtering were performed using a simple information retrieval system outlined in Fig. 2. Once the best model is determined, every tweet can receive its cluster label. The system takes a user as an input and shows the top 20 tweets from the same cluster as relevant tweets. If a user's tweets are categorised into many clusters, it displays the tweets in proportion to the number of a user's tweets in each cluster. As for indexing relevant tweets, they are ordered by the number of favourite counts as it is a good indicator of popularity.

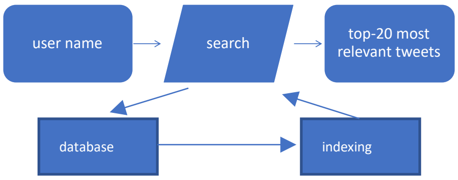

Fig. 2. Tweet filtering flow chart

IV. Results and Discussion

In this section, we present out results and discussion, which answer our research questions. Firstly, it describes the evaluation results of four models using two feature representation methods and two clustering techniques. Next, we explore each model further followed by a discussion about the quality of clusters in the best model. We then discuss methods for cluster topic interpretation using the best model. Lastly, the limitations of this study are discussed.

A. Performance evaluation and comparison of different models

Table 2 provides the optimal number of clusters obtained for each model. Additionally, a summary of silhouette scores for two feature representation methods and clustering techniques are provided in Table 3. In general, it is clear from Table 3 that doc2vec outperformed TF-IDF by 4 to 10 times more in terms of the silhouette score. In particular, doc2vec combined with k-means clustering delivered the best performance with the silhouette score of 0.4692 when there are three clusters. These results align with the observation made by a previous study conducted by [5]. Overall, our hypothesis that doc2vec can capture user interests better than TF-IDF is proven to be correct. Regarding other models, the second-best one is doc2vec combined with hierarchical agglomerative clustering. We obtained close results to the best model when the number of clusters is three or eight. However, there is no model whose silhouette score exceeded 0.1 when using the TF-IDF method regardless of clustering techniques. An observation made for the TF-IDF method is that term frequency does not work in tweet vector representation as most words only occur once in each tweet. The same discussion was made in [14]. This is reasonable considering that tweets only contain seven words on average.

TABLE II. Optimal Number of Clusters for Four Models

Feature Representation	Optimal number of clusters (k)	
	K-means	HAC[a]
TF-IDF	3	5
Doc2vec	8	3

[a] hierarchical agglomerative clustering

TABLE III. Comparison of Silhouette Score for Each Model at Three Optimal Number of Clusters

Optimal number of k	Feature Representation	Clustering	
		K-means	HAC
3	TF-IDF	0.0487	0.0394
	Doc2vec	0.4692	0.4663
5	TF-IDF	0.0517	0.0671
	Doc2vec	0.4133	0.3048
8	TF-IDF	0.0671	0.0421
	Doc2vec	0.3403	0.3048

B. Performance evaluation for individual models

In this section, we explain how the optimal number of clusters is determined for four model and provide observation about trends in evaluation results. Fig. 3 to 8 illustrate within-cluster variance and the silhouette score for each model. We only use the silhouette score for evaluation in hierarchical clustering as the elbow method is only appropriate for k-means clustering. Regarding the model with TF-IDF and k-means clustering, the elbow is not clear because there is a steady decrease in within-cluster variance over the number of clusters. Therefore, we use the silhouette score and computation time to determine the optimal number of k and we decided the value to be 3. This is because it can be seen that the silhouette score increases as the number of clusters increases. However, the computational resource also grows over clusters. This means the silhouette score and the model construction time are in a trade-off relationship. Therefore, we use optimal k equals to three so that we can maintain efficiency. The same observation about the silhouette score can be made in the model with TF-IDF and hierarchical clustering. It can be said that the silhouette score increases over the number of clusters when using TF-IDF as the feature representation method. The optimal k is determined to be five for the model using TF-IDF and hierarchical clustering as the computation time is smallest and the silhouette score shows a peak there.

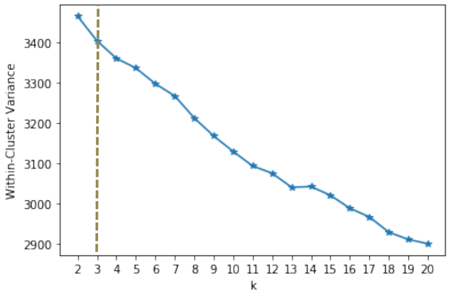

Fig. 3. Change in within-cluster variance over number of clusters for TF-IDF and k-means clustering

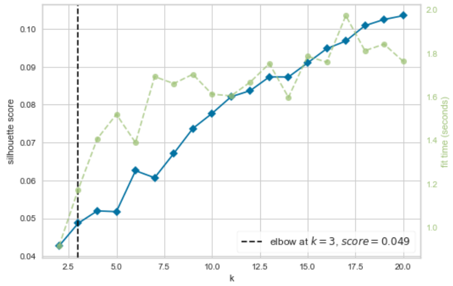

Fig. 4. Change in the silhouette score over number of clusters for TF-IDF and k-means clustering

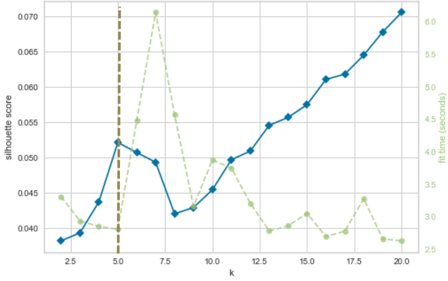

Fig. 5. Change in the silhouette score over number of clusters for TF-IDF and hierarchical agglomerative clustering

When it comes to model with doc2vec and k-means clustering, it is able to detect the elbow clearly from Fig. 6. Another interesting observation is that there is a steady decrease in the silhouette score while the number of clusters increases, which is different from TF-IDF. This is applicable for both models using doc2vec. After all, the optimal k for k-means and hierarchical clustering is determined to be eight and three respectively. The changes in the silhouette score for doc2vec models are summarised in Fig. 7 and 8.

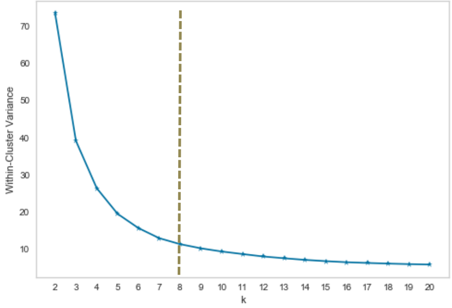

Fig. 6. Change in within-cluster variance over number of clusters for doc2vec and k-means clustering

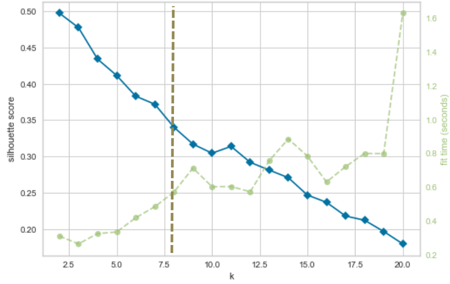

Fig. 7. Change in the silhouette score over number of clusters for doc2vec and k-means clustering

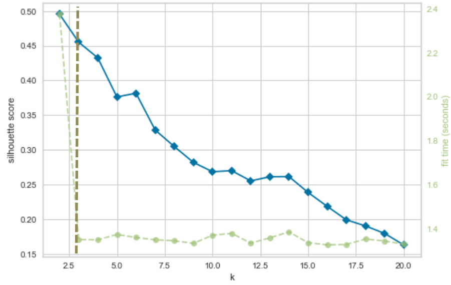

Fig. 8. Change in the silhouette score over number of clusters for doc2vec and hierarchical agglomerative clustering

C. *Analysis of cluster quality for the best model using doc2vec and k-means clustering*

In this section, we analyse the quality of clusters produced by doc2vec and k-means model using silhouette plots (Fig. 9) adopted from [18] and the Principal Component Analysis (Fig. 10). We choose number of clusters to be 8 because it is the optimal number of clusters for this model. Overall, there are some traits about good clusters from the figures. Fig. 9 shows that there is similar distribution of silhouette scores for each cluster, and every cluster has a score above the average. Moreover, the thickness of clusters is comparable except for clusters 2 and 4, which indicates tweets are evenly distributed among clusters. Fig. 10 explains the distribution of instances in reduced two dimensions. It is clear that the clusters are close to each other, but the datapoints are not scattered or mixed between clusters. Therefore, it can be said that the clusters are well separated.

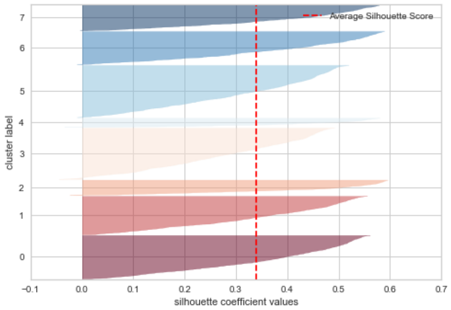

Fig. 9. Silhouette plot of doc2vec and k-means model for 3974 samples in 8 clusters

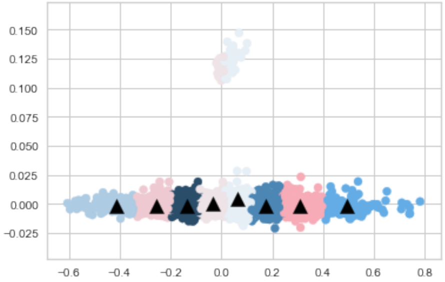

Fig. 10. PCA of doc2vec and k-means model for 3974 samples in 8 clusters

D. *Topic interpretability*

We have analysed the quality of clusters in the best model; however, the usefulness of topic clustering depends on how interpretable the cluster topics are [5]. Hence, it is necessary to examine cluster topic interpretability using the doc2vec and k-means clustering model. To address this problem, we use the top 10 frequent terms from each cluster to represent topics following [11]. This is calculated by combining all tweets in a cluster and count term frequency. The results are summaried in Table 4. We also provide word clouds which represent frequent terms in each cluster in Fig. 11. In general, it is difficult to interpret true topics in each cluster by manual inspection of the keywords. For example, there are two terms “one” and “time” in cluster 2 and if we search tweets containing those words in cluster 2, we realise that users are interested in Justin Bieber’s song called One Time. However, it is difficult to guess the true topic as those words on their own do not have much meaning. In addition, in cluster 3, there are spammy keywords such as “want”, “follower” and “retweets”, but it also contains other words such as “check” and “video”, which seem to have no relation to those spammy keywords. In summary, it is necessary to develop a better solution to reveal cluster topics when using doc2vec and k-means clustering. This is because many words on their own are not meaningful and many clusters share the same keywords.

TABLE IV. TOP-10 FREQUENT TERMS IN EACH CLUSTER

Cluster 1		**Cluster 2**		**Cluster 3**	
Topic	**TF**	**Topic**	**TF**	**Topic**	**TF**
new	105	love	46	new	139
music	104	music	45	song	69
love	56	new	31	post	58
video	43	show	29	music	50
song	39	get	27	follower	44
get	34	today	27	want	42
via	33	one	25	retweet	36
night	32	time	25	love	27
single	31	like	24	video	20
check	28	night	22	check	15
Cluster 4		**Cluster 5**		**Cluster 6**	
Topic	**TF**	**Topic**	**TF**	**Topic**	**TF**
music	81	chris	10	music	85
new	56	idol	9	new	81
may	54	kelly	9	love	65
love	52	call	9	get	44
get	47	maybe	9	check	41
feat	44	udah	8	time	33
show	42	submit	8	live	30
sign	41	keren	8	one	29
star	41	buat	8	like	29
song	40	terpilih	8	single	27
Cluster 7		**Cluster 8**			
Topic	**TF**	**Topic**	**TF**		
new	169	music	41		
music	116	love	18		
check	72	time	18		
video	58	get	16		
love	55	one	15		
song	45	new	15		
free	32	twitter	14		
download	29	night	14		
worth	28	city	13		
hanging	28	would	12		

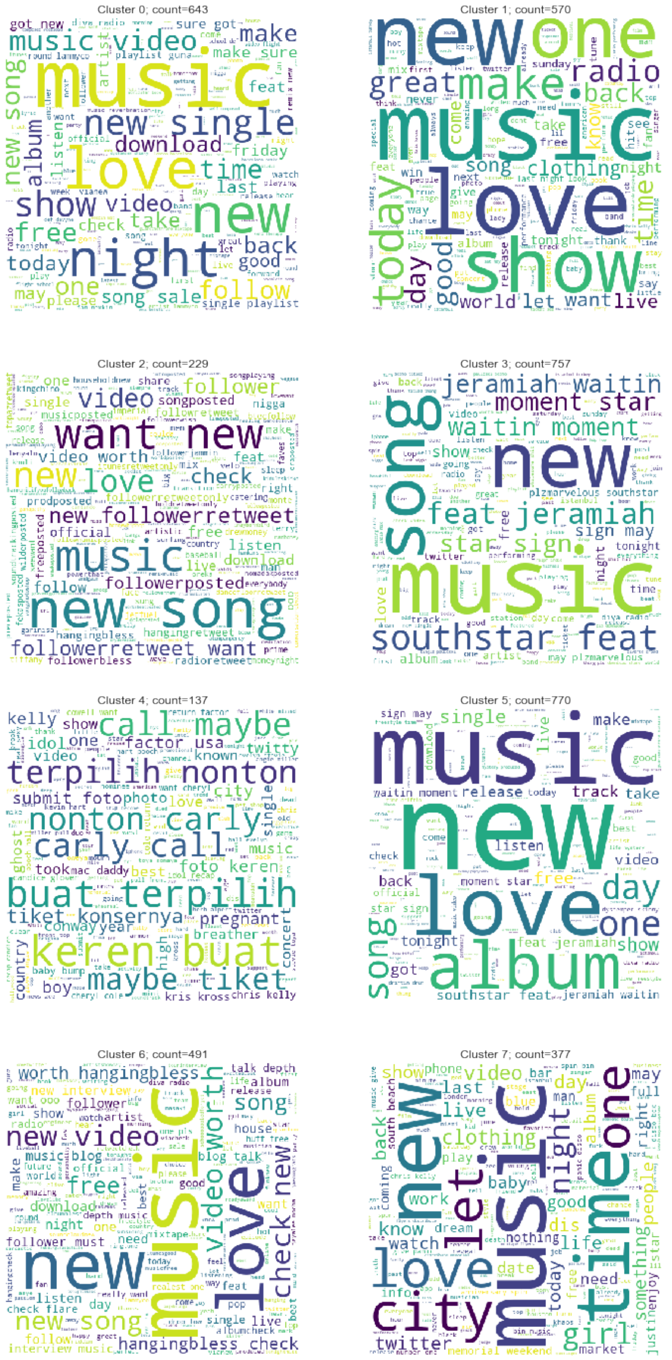

Fig. 11. Word cloud representing each cluster

E. Limitations

There are three limitations to this study. Firstly, the highest silhouette score attained from the best model was 0.4692, which is lower than 0.5. This implicates that the model can be improved to obtain a higher silhouette coefficient. A possible solution to this problem is to add a detailed pre-processing step to detect slang such as 2night and gr8. This is because we observed that a large number of slang words were discarded during the pre-processing step. Even if there is slang remaining in the corpus, the original versions of words also exist such as tonight and great, which have the same meaning as their slang equivalents. Hence, we need to detect slang and change them to their original words so that we can retain important features, and at the same time, reduce the dimensions of words. Another limitation is topic interpretability of clusters using the doc2vec and k-means clustering model. The problem comes from our approach as it is difficult to interpret true topics by manual inspection of the top 10 frequent terms. This is because each term on its own does not have an important meaning and many clusters share the same frequent terms. In order to address this issue, it is possible to use a N-gram tokeniser to detect important phrases consisting of more than 2 words such as One Time from the example in Section 4-D. Another solution is to use hashtags to substitute topic labels by identifying the top hashtags by frequency in each cluster, which was implemented in [5].

V. Conclusion

In this study we examined how to filter tweets based on user interests using two feature representation methods and two clustering techniques. Our results have demonstrated that do2vec outperformed TF-IDF in clustering evaluation. In particular, doc2vec combined with k-means clustering delivered the best performance in all cases. We also performed a top frequent term analysis to discover the topics in each cluster. The novelty of this research is that it utilised an unsupervised clustering technique, particularly the silhouette coefficient, to evaluate the models, which did not require annotated data. This aligns more with real situations as it is impossible to give labels manually to all tweets while there are 500 million posts generated daily. The significance of this research is that it is not only useful for users to find relevant tweets, but it can also help businesses to analyse what their followers are interested in, so that they can use the findings for marketing purposes. There are several improvements which can be done as future work. The first suggestion is related to the improvement in the silhouette score for the best model. We can add a detailed pre-processing step to detect slang and change them to their original forms in order to retain important features and reduce the dimension at the same time. Another enhancement can be performed to topic interpretability. The issue of using frequent terms as cluster topics is that it is difficult to interpret their true topics because those terms do not have much meaning and many clusters share the same keywords. Therefore, it is necessary to investigate further as to how to interpret cluster topics when using doc2vec and k-means clustering.

References

[1] Albishre, K., Li, Y., Xu, Y., & Huang, W. (2019). Query-based unsupervised learning for improving social media search. *World Wide Web*, *23*(3), 1791–1809. https://doi.org/10.1007/s11280-019-00747-0

[2] Alnajran, N., Crockett, K., McLean, D., & Latham, A. (2017). Cluster Analysis of Twitter Data: A Review of Algorithms. *Proceedings of the 9th International Conference on Agents and Artificial Intelligence*, 239–249. https://doi.org/10.5220/0006202802390249

[3] Alvarez-Melis, D., & Saveski, M. (2016). Topic Modeling in Twitter: Aggregating Tweets by Conversations. *ICWSM*.

[4] Chen, J., Nairn, R., Nelson, L., Bernstein, M., & Chi, E. (2010). Short and tweet: experiments on recommending content from information streams. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, *2*, 1185–1194. https://doi.org/10.1145/1753326.1753503

[5] 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

[6] Dutta, S., Ghatak, S., Das, A., Gupta, M., & Dasgupta, S. (2019). Feature selection-based clustering on micro-blogging data. *Advances in Intelligent Systems and Computing*, *711*, 885–895. https://doi.org/10.1007/978-981-10-8055-5_78

[7] Ferrara, E., Jafariasbagh, M., Varol, O., Qazvinian, V., Menczer, F., & Flammini, A. (2013). Clustering memes in social media. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 548–555. https://doi.org/10.1145/2492517.2492530

[8] 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.

[9] Ibtihel, B., Lobna, H., & Maher, B. (2018). A Semantic Approach for Tweet Categorization. *Procedia Computer Science*, *126*, 335–344. https://doi.org/10.1016/j.procs.2018.07.267

[10] 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

[11] Jun, S., Park, S., & Jang, D. (2014). Document clustering method using dimension reduction and support vector clustering to overcome sparseness. *Expert Systems with Applications, 41*(7), 3204–3212. https://doi.org/10.1016/j.eswa.2013.11.018

[12] Klinczak, M., & Kaestner, C. (2016). Comparison of clustering algorithms for the identification of topics on twitter. *Latin American Journal of Computing - LAJC, 3*, 19–26.

[13] Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. Proceedings of the 31th international conference on machine learning, ICML 2014, Beijing, China, 21–26 June 20141188–1196.

[14] Li, Z., Zhou, D., Juan, Y., & Han, J. (2010). Keyword extraction for social snippets. *Proceedings of the 19th International Conference on World Wide Web*, 1143–1144. https://doi.org/10.1145/1772690.1772845

[15] Nanas, N., Uren, V., & de Roeck, A. (2004). A comparative evaluation of term weighting methods for information filtering. *Proceedings. 15th International Workshop on Database and Expert Systems Applications, 2004*, 13–17. https://doi.org/10.1109/DEXA.2004.1333442

[16] Ounis, I., Macdonald, C., & Lin, J. (2013, August). *Overview of the trec-2011 microblog track.* NIST. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=914332

[17] Patki, U., & Khot, P. (2017). A Literature Review on Text Document Clustering Algorithms used in Text Mining. *Journal of Engineering Computers & Applied Sciences, 6*, 16-20.

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

[19] Salton, G & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management 24* (5): 513-523.

[20] Sedhai, S., & Sun, A. (2015). HSpam14: A Collection of 14 Million Tweets for Hashtag-Oriented Spam Research. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 223–232. https://doi.org/10.1145/2766462.2767701

[21] Zhang, S., Zhang, S., Yen, N., & Zhu, G. (2017). The Recommendation System of Micro-Blog Topic Based on User Clustering. *Mobile Networks and Applications*, *22*(2), 228–239. https://doi.org/10.1007/s11036-016-0790-9