Multi-cluster words in Model-Based Clustering of Short Text Streams

By

Bellamkonda Sachin

17MA20008

Under the supervision of

Dr.Pawan Kumar

Department of Mathematics

Indian Institute of Technology Kharagpur

Certificate

This is to certify that the work contained in this project entitled, “Disambiguating Multi-Cluster words in Model-Based clustering of short text streams” submitted by Bellamkonda Sachin-17MA20008 to Indian Institute of Technology, India towards partial requirement of “Masters of Science” in Mathematics and Computing is a record of Bona fide project carried out by him under my super vision and guidance is worthy of consideration.

Table of Contents

Abstract

The importance of short text stream clustering has increased due to major growth of short text in social media. Here we discuss a streaming algorithm that works on short text streams, so unlike normal text streams, the problem with short text streams is that the frequency of words occurring is less, which faces the sparsity problem. Many Classification models like Dynamic Topic Model (DTM), Topic Tracking Model (TTM) etc. are proposed, but these models assume that each document has rich contents, most of these don't work on short text streams and also most of them assume a fixed number of clusters so they can't deal with the topic drift property efficiently and also the problem arrived due to the common words in different clusters (multi-Cluster words). Our results have shown that the MStream algorithm with removing the common cluster words has given us better results than MStream algorithm.

Introduction

Short text streams emerging on the internet have become more and more popular in recent times. The task on topic modelling these short text streams from the documents incoming has many applications such as text summarization, recommendation engines, google search snippets etc.

The problem with general streaming data is that the data will be coming continuously to our model, so storing the documents and iterating them again and again is out of equation. Each document may come at a one-time which is one pass scheme. Also, more generally they come in batches which is the batch scheme. Most of the data coming will be in batches and new clusters are evolving from this continuous inflow of batches. The problem of short text streams over large text data documents is that the words in short texts appear less frequently this led to data sparsity problem. Since this is streaming data there can be new topics evolving over the period of time. Therefore, the documents coming may not set into the clusters that are already created from the previous documents, this is called concept drift problem. This concept drift problem makes the prediction less accurate as time progresses because of the change in distribution of data over the time. And also, when the clusters are created there can be some homograph words in the clusters that are already belong to different cluster, but due to these words there is a probability that a new document may go into the existing cluster although it belongs to another new cluster.

So, we are going to work on an algorithm introduced to tackle the above-mentioned problems. First, we introduce a MStream clustering Algorithm works completely fine with both the one pass scheme and batch scheme. With experimental results shown that even with the one pass scheme MStream algorithm is getting good results and with the batch scheme as we can iterate every batch multiple times until next batch arrives, we can get even better results compared to one pass scheme. Instead of giving the probability of each document belong to different topics we rather assign each document belongs to one topic. By this way we can handle data sparsity problem in incoming data.

The number of clusters are not fixed before running the model so, the data drift problem is solved because when the new topic arises the document can go into newly created clusters. As in the daily life application the twitter data (short text data) coming will increase day by day so running time to cluster these documents and the space used to store these documents will increase exponentially and also, we are interested in topics of specific period of time. So, for this we introduce a new algorithm called MStreamF algorithm which solve these problems. We will build this on MStream algorithm with removing old texts which is done by removing old clusters in the previous batches. We will remove the outdated clusters by having a fixed number of batches (fixed before running the model) and these batch of batches need to be stored for the estimation of new batches when number of batches exceeds this threshold value, we remove those clusters that are associated with old batches that is we will be removing the information from the cluster feature.

In this model to handle the problem of multi cluster words existing in the clusters we remove the multi cluster words with high entropy which can reduce the topic ambiguity problem.

Related Work

Text stream data clustering, we can classify them as shown below.

1.Similarity Based Stream Clustering

This type of method mainly chooses vector-based model to find the similarity between the documents by representing documents as vectors and finding similarity between them using cosine similarity. For example, CluStream[10] based on the concept of micro clusters. Micro clusters are data structures which summarize a set of instances from the stream. CluStream has both online and offline phase and uses pyramidal time frame.

The drawback of this type of clustering is that there is a threshold value that needs to be fixed for the documents to go into new cluster or join the existing cluster. Unlike this in our method we do not fix any threshold for our documents to go into existing cluster or not, instead in the proposed method it calculates probabilities whether a new cluster need to be created fir the document or it can fit into the existing one. By following this method, we can overcome the concept drift problem in the streaming data.

2.Model Based Stream Clustering

Model based clustering considers the data are generated by a mixture of underlying probability distributions. Differently from k-means, in model-based clustering each document has a probability to which cluster the document needs to be assigned.

There are many different methods like Latent Dirichlet Allocation (LDA) [11], streaming LDA(ST-LDA) [12], topic tracking model (TTM) [13]. But these methods do not handle well for the short text streams since the information in document is not enough for the document multinomial distributions in clusters There is a method Dynamic Clustering Topic Model (DCT) which can be used for short text streams in which each document coming is assigned to single topic unlike the probabilistic to each and every cluster.

In the traditional methods we can see that the number of topics (or) clusters are fixed from starting which is not true in case of text streams where we have the concept drift problem and the short text problem.

Approach

We have divided this into three parts initially we introduce how the documents, clusters are shown in this model and then we implement MStream algorithm which works with data sparsity and data drift problem was extended to MStream Algorithm with some rules to remove the outdated clusters in the cluster feature.

Representation

In general similarity based clustering methods keep term frequency-inverse document frequency (TF-IDF) which are used for estimating the corresponding weights to the words that are in the clusters. Differently, in this method each document is a collection of words from the streaming data and the frequencies for each of the words that are present in the document and the other difference is that in our representation we assume documents are from a multinomial distribution.

Here to represent a document a cluster feature is used. Here cluster feature is tuple with three characters co-occurrences of the words in the cluster z, number of documents that are present in each of the cluster (m_z) and number of words (words from the documents) in each cluster z(n_z). The cluster feature is very useful when a new document is updated into the cluster or deleted from the cluster, we just need to update the above three variables.

MStream Algorithm

This is a model based clustering algorithm. This can work for one pass scheme and updated version of this can work for the batch process as well and the results will get even better after multiple iterations of the batches until the next batch arrives.

Establishing the relation between created clusters and documents is the main thing in the clustering of documents. General Similarity based clusters the concept drift problem is solved using a threshold value. So, when a new document arrives it checks the similarity with all the existing clusters if it exceeds the threshold value then a new cluster is formed and this document is allocated to it. It is difficult to set a threshold value for all the documents coming manually. Differently in our model our documents are coming from Dirichlet Process Multinational Mixture (DPMM). Based on DPMM we calculate the probability of document that goes to existing cluster is given by:

$$p(z_d = z|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta)$$

$$\propto \frac{m_{z,\neg d}}{D - 1 + \alpha D} \frac{\prod_{w \in d} \prod_{j=1}^{N_d^w} (n_{z,\neg d}^{w} + \beta + j - 1)}{\prod_{i=1}^{N_d} (n_{z,\neg d} + V\beta + i - 1)}$$

Here, ¬d means from the cluster feature this document was removed, which can be used in batch process of the MStream algorithm. For a new coming document ¬d does not influence the CF vectors. Different from static text clustering, D is count of documents and V is number of unique vocabularies recorded.

And the probability that the document goes to the new document is,

$$p(z_d = K + 1|\vec{z}_{\neg d}, \vec{d}, \alpha, \beta)$$

$$\propto \frac{\alpha D}{D - 1 + \alpha D} \frac{\prod_{w \in d} \prod_{j=1}^{N_d^w} (\beta + j - 1)}{\prod_{i=1}^{N_d} (V\beta + i - 1)}$$

αD is the pseudo measure of number of documents in the cluster, β is the pseudo count of each word that are in the cluster.

From the above two equations we can find that in the left part of the equation the probability is directly proportional to number of documents that are in the cluster. That means if a document is incoming to our model it tries go into the cluster that has more documents by using this we handle the case that we don't have infinite number of clusters that we don't have to check probability with. The second part of above two equations is similarity of document with the clusters with frequency of words that are present in cluster if cluster shares a greater number of common words with the document, then it goes into the cluster.

By combining both the effects the documents decide to which cluster it should go based on the probabilities calculated.

For the one-pass scheme documents are coming one by one. So, when the first document arrived a new cluster will be created and the cluster feature will be updated for this document and when the next document comes it calculates the above two probabilities and if the probability is higher for creating a new cluster, then a new cluster is created and a tuple for cluster matrix is updated or it goes to existing cluster then we update that cluster feature.

For the batch-scheme documents comes in batches the same process is done and when one time iteration done on this batch from the second iteration onwards the document that is already assigned will be removed from cluster feature and again the probabilities are calculated for this document and re-assigned to the existing or a new cluster in this, we give equal priorities for each document in the batch by iterating more than once and this is how concept drift problem is managed.

MStreamF Algorithm

As the documents will come continuously the storage and running time of algorithm increases continuously which is not suitable. Normally we are interested only in the current topic. So, instead of keeping all the documents we remove the outdated batches. Deletion means when batch b is removed all the documents of the batch b are removed from the cluster feature this proposed algorithm is MStreamF algorithm.

In this we have a parameter B_s which keeps information of how many past batches we need to keep when the number of batches iterated is greater than B_s we remove all the old batch documents from the clusters and for the new batch we apply MStream to cluster them. There maybe a chance that after deleting outdated batches some clusters maybe empty we need to make sure that upcoming document choosing that cluster is not recommended. That means when a cluster becomes empty no document goes into that cluster again.

4 Experimental Setup

4.1 Data

We are using short real news data, tweet data and the other two which are variations of the news and tweet data which are sorted on the topics.

- **Tweets** – This data is taken from TREC 2011-2015 microblog and this contains 30,322 rows of tweets relevant 269 queries.
- **News** – The data contains 11,109 news headlines which are classified into 152 clusters. Average length of text is 6.23
- **News-T and Tweets-T** – In practical case we observe that a topic appear over a period of time and vanish again so News-T and Tweets-T are sorted with respect to the topics in News and Tweets.

Data Source: https://trec.nist.gov/data/microblog.html

4.2 Evaluation Matrix

Normalized Mutual Information (NMI) one of the measures for evaluating the performance of clusters created when ground truth values of documents are known. It calculates the statistical details common to the predicted clusters for the document and ground truth clusters for each document.

$$NMI = \frac{\sum_{c,k} n_{c,k} \log\left(\frac{N \cdot n_{c,k}}{n_c \cdot n_k}\right)}{\sqrt{\left(\sum_c n_c \log \frac{n_c}{N}\right)\left(\sum_k n_k \log \frac{n_k}{N}\right)}}$$

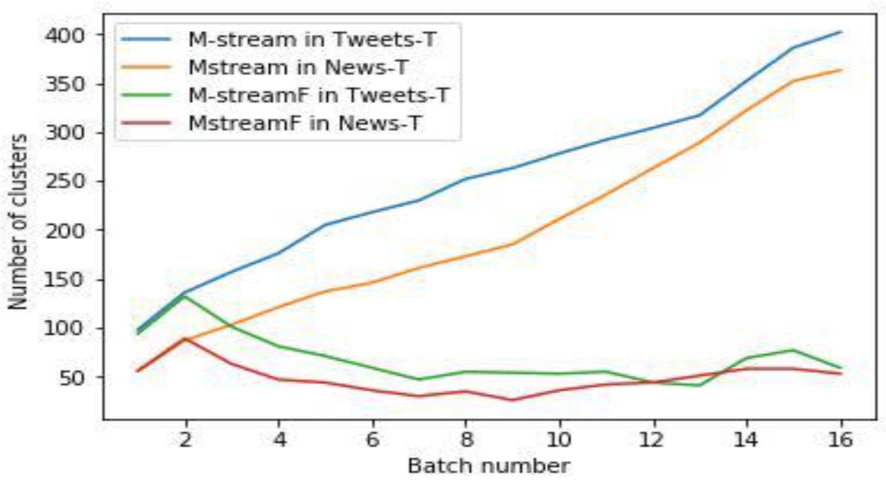
N is the total number of documents, n_k is the number of documents in cluster k, n_c is the number of documents in class c, The range of NMI is between 0 and 1. I f NMI value is 1 then all the predictions made are same as ground truth clusters. NMI value will near 0 if the clusters are unevenly distributed.

Results

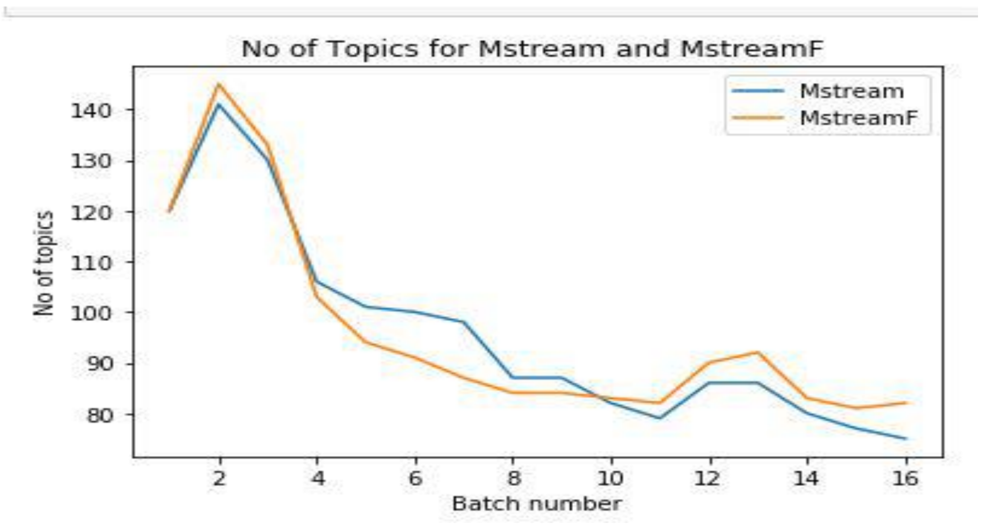
From the below table we can observe that we have obtained a good result on MStream and MStreamF algorithms and we can also observe that the MStreamF algorithm performs with higher accuracy on Tweets-T and News-T means that removing outdated clusters works better in case of News-T and Tweets-T since in these data sets the data is sorted in the order of topics evolved so forgetting rules worked well also HMStream and HMStreamF works better compared to MStream and MStreamF.

	MStream	MStreamF	HMStream	HMStreamF
News	0.7712	0.7647	0.8529	0.7612
Tweets	0.8313	0.8008	0.8548	0.8278
News-T	0.8742	0.8758	0.8749	0.8816
Tweets-T	0.8370	0.8809	0.8745	0.9036

Below is the graph on number of clusters vs batch number on all News-T, Tweets-T datasets in MStream and MStreamF

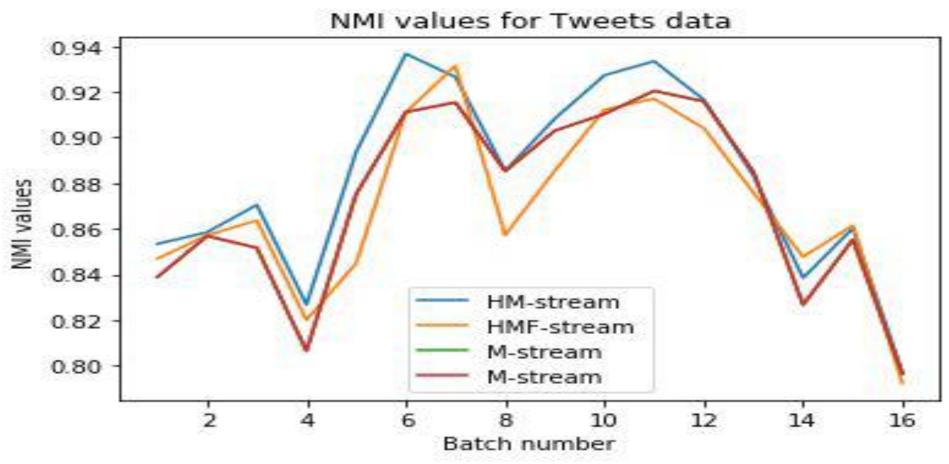


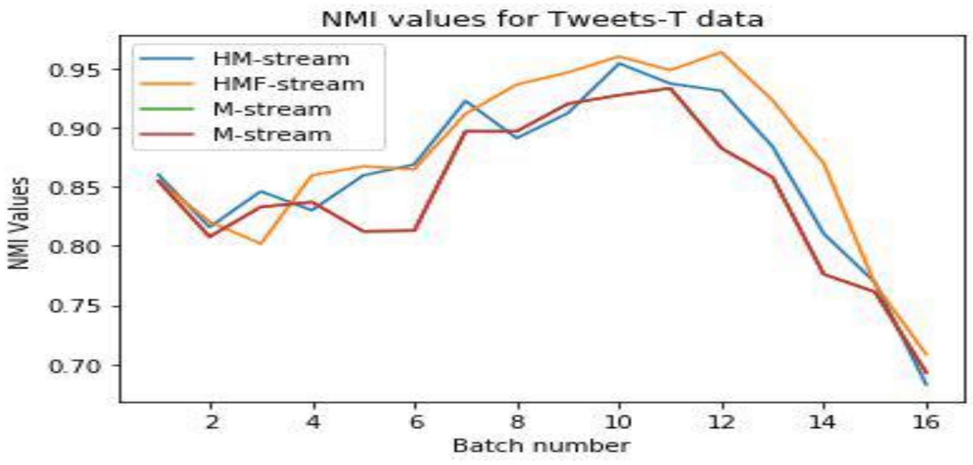
Below is the graph for number of clusters vs batch number on News data in MStream and MStreamF



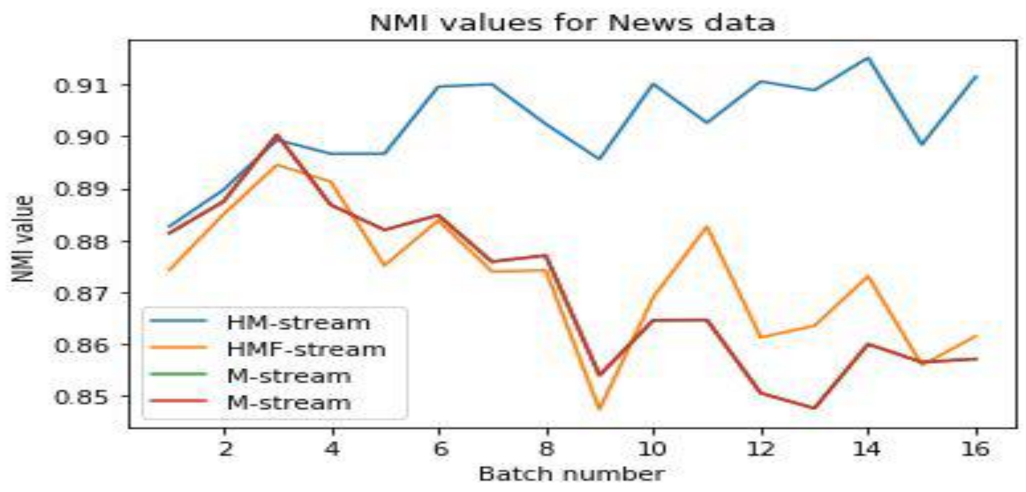
As we can see from the above two graphs that number of clusters initially are same for both MStream and MStreamF but as moving forward the number of clusters in MStreamF are less compared to MStream because of the forgetting rules we introduced in MStreamF.

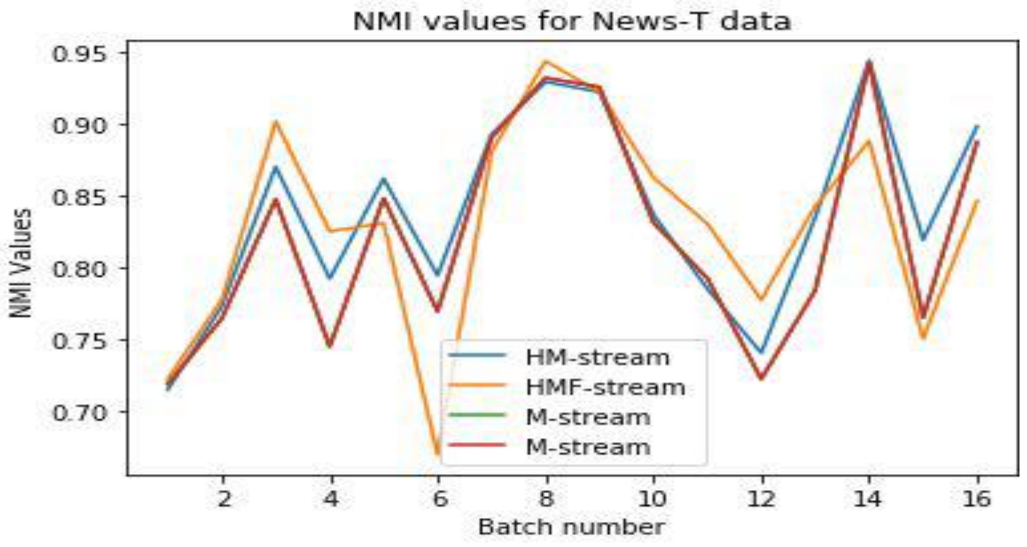
Below is a graph is between the NMI scores vs Batch numbers on Tweets data and Tweets-T respectively in all MStream, MStreamF, HMStream, HMStreamF.





Similarly, below are the plots between NMI vs Batch number for News and News-T respectively.

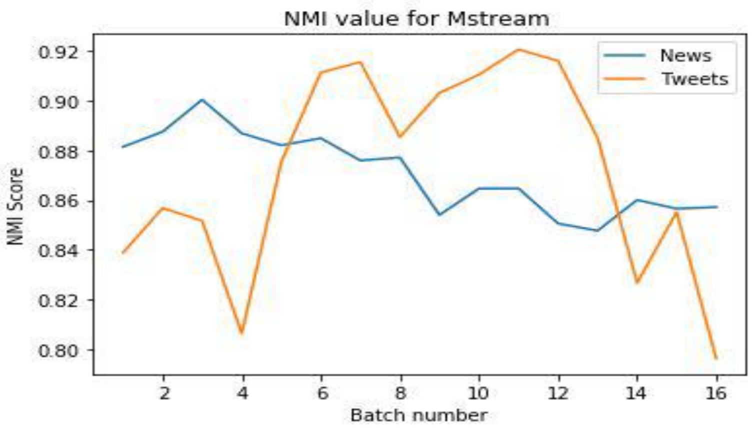


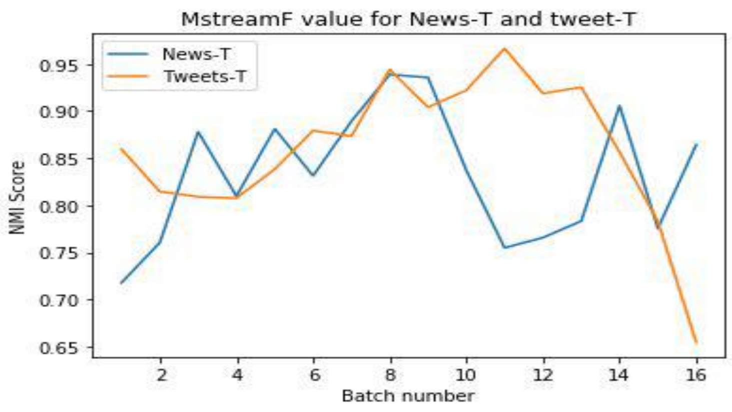


From the above four figures we can observe that NMI values in each stream of data is not uniform and fluctuates. This shows us that there are many independent cases in batches which shows us MStream and MStreamF can adapt to many situations.

From these figures we can also observe that the performance of updates MStream and MStreamF algos is better because we have removed the multi cluster words with high entropy values the performance increased.

Below we applied MStream on Tweets and News datasets. MStreamF on Tweets-T and News-T datasets.





Conclusions

We have introduced MStream and MStreamF algorithms for clustering of short text streams which are able to handle data sparsity problem and concept drift problem in short text streams. From the MStream algorithm we have updated it to MStreamF by removing old batch documents in clusters and we got better results from these in News-T and Tweets-T datasets. And we have removed multi-cluster words with high entropy, by removing them we have achieved better results compared to MStream and MStreamF.

References

[1] Yin, J., Chao, D., Liu, Z., Zhang, W., Yu, X. and Wang, J., 2018, July. Model-based clustering of short text streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2634-2642).
[2]Cheng, X., Yan, X., Lan, Y. and Guo, J., 2014. Btm: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering* , *26* (12), pp.2928-2941.
[3]Phan, X.H., Nguyen, C.T., Le, D.T., Nguyen, L.M., Horiguchi, S. and Ha, Q.T., 2010. A hidden topic-based framework toward building applications with short web documents. *IEEE Transactions on Knowledge and Data Engineering* , *23* (7), pp.961-976.
[4]Li, P., He, L., Wang, H., Hu, X., Zhang, Y., Li, L. and Wu, X., 2017. Learning from short text streams with topic drifts. *IEEE transactions on cybernetics* , *48* (9), pp.2697-2711.
[5]Rakib, M.R.H., Zeh, N. and Milios, E., 2020, September. Short Text Stream Clustering via Frequent Word Pairs and Reassignment of Outliers to Clusters. In *Proceedings of the ACM Symposium on Document Engineering 2020* (pp. 1-4).
[6] Paul, M.J. and Dredze, M., 2014. Discovering health topics in social media using topic models. *PloS one* , *9* (8), p.e103408.
[7]Resch, B., Usländer, F. and Havas, C., 2018. Combining machine-learning topic models and spatiotemporal analysis of social media data for disaster footprint and damage assessment. *Cartography and Geographic Information Science* , *45* (4), pp.362-376.
[8]Guo, X., Xiang, Y., Chen, Q., Huang, Z. and Hao, Y., 2013. LDA-based online topic detection using tensor factorization. *Journal of Information Science* , *39* (4), pp.459-469.
[9]Yan, X., Guo, J., Lan, Y., Xu, J. and Cheng, X., 2015, January. A probabilistic model for bursty topic discovery in microblogs. In *AAAI* (pp. 353-359).
[10] Aggarwal, C.C., Philip, S.Y., Han, J. and Wang, J., 2003, January. A framework for clustering evolving data streams. In *Proceedings 2003 VLDB conference* (pp. 81-92). Morgan Kaufmann. .
[11] Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. *Journal of machine Learning research* , *3* (Jan), pp.993-1022.
[12] Amoualian, H., Clausel, M., Gaussier, E. and Amini, M.R., 2016, August. Streaming-lda: A copula-based approach to modeling topic dependencies in document streams. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 695-704). .
[13] Iwata, T., Watanabe, S., Yamada, T. and Ueda, N., 2009, June. Topic tracking model for analyzing consumer purchase behavior. In *Twenty-First International Joint Conference on Artificial Intelligence* .