Short Text Clustering in Continuous Time Using Stacked Dirichlet-Hawkes Process with Inverse Cluster Frequency Prior

Avirup Saha
avirupsaha@iitkgp.ac.in
IIT Kharagpur
Kharagpur, India

Balaji Ganesan
bganesa1@in.ibm.com
IBM Research
Bengaluru, India

ABSTRACT

Traditional models for short text clustering ignore the time information associated with the text documents. However, existing works have shown that temporal characteristics of streaming documents are significant features for clustering. In this paper we propose a stacked Dirichlet-Hawkes process with inverse cluster frequency prior as a simple but effective solution for the task of short text clustering using temporal features in continuous time. Based on the classical formulation of the Dirichlet-Hawkes process, our model provides an elegant, theoretically grounded and interpretable solution while performing at par with recent state of the art models in short text clustering.

ACM Reference Format:
Avirup Saha and Balaji Ganesan. 2021. Short Text Clustering in Continuous Time Using Stacked Dirichlet-Hawkes Process with Inverse Cluster Frequency Prior. In *MileTS '21: 7th KDD Workshop on Mining and Learning from Time Series, August 14th, 2021, Singapore.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In recent years, with the proliferation of social media such as Facebook or Twitter, a massive volume of short text data is being continuously generated on social media and online news platforms. Clustering of such short text documents has a huge potential impact in identifying trending topics, extracting related information pertaining to any given topic, user-specific recommendation etc. This has led to several efforts in the recent past to design algorithms that can cluster short text documents using variants of the Dirichlet Process combined with heuristic techniques [8, 13, 18, 25–27]. There are also some works that attempt to cluster the documents using temporal information (in continuous time) associated with streaming short text documents, most notably Dirichlet-Hawkes Process (DHP) [10] and its extensions [16, 19].

Du *et al.* [10] argue that besides textual information, temporal information is also significant for the clustering of online document streams. For example, when a catastrophic event occurs, news of the event is first published from preliminary reports, followed by rapid circulation of follow-up articles. After a while as the influence of the event passes, the rate of generation of articles dies down. These self-excitation phenomena can be modeled using the Hawkes process [12] in continuous time. Also, articles relevant to different types of news stories can exhibit heterogeneous temporal dynamics, which may be a significant feature for clustering. Such characteristics are also observed in microblogs [4]. Despite these advantages, DHP [10], although a clean and robust model, suffers from performance limitations in comparison to the state-of-the-art approaches [8, 18] Hence in this paper, we propose some improvements to the classical DHP which give it comparable or better performance to these state-of-the-art models.

First, we observe that DHP assumes a uniform prior over the word distribution in the corpus which signifies no prior knowledge of the relative importance of the words in the vocabulary. However, this naïve assumption of a uniform prior can be improved by incorporating the *Inverse Cluster Frequency (ICF)* information. Since the term-frequency (tf) information is inherently captured by the Dirichlet process, we posit that incorporation of the normalized ICF as a prior is a natural extension to the Dirichlet process which can be related to the well-known tf-idf paradigm as ICF gives more importance to words that occur in only a few clusters than words which occur across many clusters.

Secondly, when the ground truth clusters are temporally sparse, DHP often creates multiple clusters for the same ground truth topic due to the Hawkes process being an imperfect representation of the temporal dynamics of such clusters. This motivates us to create a stacked version of the DHP by inducing a second order clustering using the Dirichlet process to merge textually similar clusters.

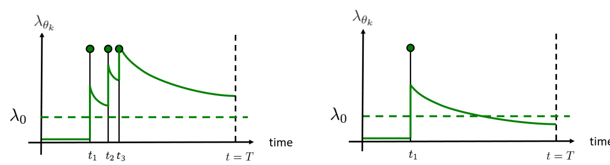
To summarize, we propose two intuitive, explainable and theoretically sound modifications to the basic DHP, viz. (i) using ICF prior and (ii) a stacked version with a second order clustering. We show that our simple and intuitive techniques achieve comparable results to the state-of-the-art approaches in short text clustering.

2 RELATED WORK

Text clustering is an important problem for NLP as it is closely related to topic modeling. Hence several approaches have been proposed in NLP literature, as detailed by surveys [1, 15, 17, 20]. The earliest well-known model for text clustering was LDA [6]. It was subsequently extended by variants for stream clustering [2, 3, 5, 21, 22, 28]. Specialized model-based approaches for short text clustering began with GSDMM [25], followed by DCT [14], GSDPMM [26], FGSDMM+ [27]. Streaming aspects of short text clustering such as cluster evolution were addressed by MStream and MStreamF which introduced a forgetting mechanism for old clusters [24]. A word embedding based approach was considered in NPMM [7]. OSDM [13] introduced a fully online model with

MileTS '21, August 14th, 2021, Singapore

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456



(a) Dense cluster **(b) Sparse cluster**

Figure 1: Intensity distributions of Hawkes processes corresponding to a dense and a sparse cluster. For the sparse cluster, the cluster intensity $\lambda_{\theta_k}(t)$ at $t = T$ falls below the base intensity λ_0.

word-to-word cooccurrence information. A biterm-based dirichlet multinomial model was proposed by [8] to address the sparsity of co-occurrence information in short documents. Finally, Rakib *et al.* [18] demonstrate the use of a combination of heuristic techniques, such as considering frequent word pairs and outlier reassignment to achieve strong results.

None of the above approaches consider the documents to arrive in continuous time and hence ignore the temporal features of the documents. In contrast, the Dirichlet-Hawkes Processs (DHP) [10] was proposed to cluster streaming documents in continuous time. It was followed by hierarchical variants, such as HDHP [16] and HDGMHP [19]. Xu and Zha [23] proposed a Dirichlet mixture model of Hawkes processes in the general context of event sequence clustering, where the events need not contain textual information. In this paper, we base our models on the version of DHP proposed by [10].

Ding *et al.* [9] proposed a semi-supervised Dirichlet-Hawkes process for topic tracking in Twitter with Hashtag Supervision and Relevance Kernel Supervision. Relevance Kernel Supervision discounts the effect of common words using a TF-IDF related measure (BM25). However, this is quite different from the ICF prior we have considered in this paper. To be specific, the ICF prior is based on the cluster frequency of words and not on the document frequency. So a word which occurs often in a cluster but not across all documents may have a higher impact than a word which occurs often across the whole set of documents but rarely in that cluster. Apart from this, our approach differs from [9] in that (i) our approach is unsupervised while [9] is semi-supervised and (ii) our approach is domain-agnostic while [9] has been specifically designed for Twitter hashtags.

3 PROPOSED APPROACH

In this section we first describe DHP as proposed by [10] with its limitations, and then describe the modifications we make to address them.

3.1 DHP

Slightly modifying the notation of [10], let the incoming documents be denoted by $d_{1:n}$ and let $s_{1:n}$ and $t_{1:n}$ be the latent cluster indicator variables and document times for these n documents. We assume $t_i < t_{i+i} \forall i \in [1 : n-1]$. Corresponding to these documents DHP generates a series of samples $\theta^d_{1:n}$ where each distinct value of θ^d_i represents a cluster.

If at time t_n there are K distinct values $\theta_{1:K}$ of $\theta^d_{1:n}$, then $s_n \in \{1, 2, \ldots K, K+1\}$ where $s_n = K + 1$ denotes a new cluster and $0 < s_n \leq K$ denotes an existing cluster. Let the uniform prior θ_0 be a V dimensional vector (where V denotes the vocabulary size) where every element is a constant value, say 0.01. We obtain the posterior likelihood $P(s_n|d_n, t_n, \text{rest}) \sim P(d_n|s_n, \text{rest})P(s_n|t_n, \text{rest})$, where $P(d_n|s_n, \text{rest})$ is given by

$$\frac{\Gamma(C^{s_n} + \sum_v^V \theta_0[v]) \prod_v^V \Gamma(C_v^{s_n} + C_v^{d_n} + \theta_0[v])}{\Gamma(C^{s_n} + C^{d_n} + \sum_v^V \theta_0[v]) \prod_v^V \Gamma(C_v^{s_n} + \theta_0[v])}$$

if $0 < s_n \leq K$, and

$$\frac{\Gamma(\sum_v^V \theta_0[v]) \prod_v^V \Gamma(C_v^{d_n} + \theta_0[v])}{\Gamma(C^{d_n} + \sum_v^V \theta_0[v]) \prod_v^V \Gamma(\theta_0[v])}$$

if $s_n = K + 1$. Here C^{s_n} is the total word count of cluster s_n, C^{d_n} is the total word count of document d_n, and $C_v^{s_n}$ and $C_v^{d_n}$ are the corresponding counts of the vth word.

$P(s_n = k|t_n, \text{rest})$ is given by

$$\begin{cases} \frac{\lambda_{\theta_k}(t_n)}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i^d}(t_n, t_i)} & 0 < k \leq K \\ \frac{\lambda_0}{\lambda_0 + \sum_{i=1}^{n-1} \gamma_{\theta_i^d}(t_n, t_i)} & k = K + 1 \end{cases}$$

where λ_0 is the base intensity of a background Poisson process, λ_{θ_k} is the intensity of the Hawkes process corresponding to the kth cluster, and $\gamma_{\theta_i^d}(t_n, t_i) = \exp(-|t_n - t_i|)$. See [10] for the expression of λ_{θ_k}. Using these probabilities, Sequential Monte Carlo sampling is used to infer the cluster label of each document.

Limitations: The above formulation has two limitations:

(1) The uniform prior θ_0 gives equal importance to all words. Rakib *et al.* [18] have observed that words that occur across many clusters create noise which may mislead the clustering algorithm.
(2) While the Hawkes process very accurately describes the temporal dynamics of dense clusters (Fig. 1a), it is a poor fit for temporally sparse clusters (Fig. 1b). Let us suppose that a new document comes at time $t = T$ which has the same word distribution as the sparse (singleton) cluster of Fig. 1b. Then, for the above sampling procedure even though the textual probability $P(d_n|s_n, \text{rest})$ will be maximum for this cluster, the temporal probability $P(s_n|t_n, \text{rest})$ will be higher for the new cluster $s_n = K + 1$ since $\lambda_0 > \lambda_{\theta_k}$. If $\lambda_0 - \lambda_{\theta_k}$ be sufficiently high, this document will be assigned to a new cluster. Decreasing λ_0 is not a solution since this causes large clusters with high λ_{θ_k} to absorb irrelevant documents, yielding less homogeneity (we verify this experimentally in Figure 2). Also, tuning λ_0 may be difficult in practice.

3.2 DHP with ICF prior

To address the first limitation of DHP, instead of θ_0, for each incoming document d_n we compute the ICF prior θ_n^{ICF} as a V-dimensional vector. We first compute the ICF for every word at time t_n as

$$ICF^n[v] = \log\left(\frac{K}{\text{\# clusters containing } v\text{th word}}\right)$$

where we have used log weighting for the ICF[1]. Then we obtain the normalized prior θ_n^{ICF} as

$$\theta_n^{ICF}[v] = \frac{ICF^n[v]}{\sum_v ICF^n[v]}$$

3.3 Stacked DHP with ICF prior

To address the second limitation of DHP, we propose to perform a second-order clustering[2] by using a Dirichlet process with ICF prior on the clusters obtained by DHP, with the aim of merging clusters of the same topic. Let us assume the documents $d_{1:n}$ have been grouped into K clusters $c_{1:K}$ by DHP. Our task is to produce a series of second-order cluster labels $s_{1:K}^2$ for these K clusters. We process the clusters sequentially in the order in which they were generated by DHP in discrete timesteps. Let c_m be the first order cluster to be processed in the mth time step. If there are L second order clusters created by the Dirichlet process at this time, then $s_m^2 \in \{1, 2, \ldots L, L+1\}$ where $s_m^2 = L + 1$ denotes a new cluster.

The posterior likelihood $P(s_m^2|c_m, \text{rest}) \sim P(c_m|s_m^2, \text{rest})P(s_m^2|\text{rest}) \sim P(c_m|s_m^2, \text{rest})$ (assuming $P(s_m^2|\text{rest})$ is constant). It is given by

$$\frac{\Gamma(C^{s_m^2} + \sum_v^V \theta_m^{ICF2}[v]) \prod_v^V \Gamma(C_v^{s_m^2} + C_v^{c_m} + \theta_m^{ICF2}[v])}{\Gamma(C^{s_m^2} + C^{c_m} + \sum_v^V \theta_m^{ICF2}[v]) \prod_v^V \Gamma(C_v^{s_m^2} + \theta_m^{ICF2}[v])}$$

if $0 < s_m^2 \leq L$, and

$$\frac{\Gamma(\sum_v^V \theta_m^{ICF2}[v]) \prod_v^V \Gamma(C_v^{c_m} + \sum_v^V \theta_m^{ICF2}[v])}{\Gamma(C^{c_m} + \sum_v^V \theta_m^{ICF2}) \prod_v^V \Gamma(\theta_m^{ICF2}[v])}$$

if $s_m^2 = L + 1$. Here θ_m^{ICF2} is the second order inverse cluster frequency prior, computed in the same manner as θ_m^{ICF} on the second-order clusters but without the log weighting of the ICF [3]. Using these probabilities, the second order cluster label of each cluster is obtained by sampling from a multinomial distribution and accordingly we obtain the new set of clusters.

4 EXPERIMENTS

In this section we describe the datasets used, the baselines, the evaluation metrics, and the experimental results.

Dataset	Clusters	Docs	Avg. Len	Avg. Clus-TD
TREC	269	25868	8.28	27.2
uci_news	448	10348	6.61	5806.5

Table 1: Dataset statistics. Avg. Len refers to the average document length in words. Avg. Clus-TD refers to average temporal density in events/hr of the clusters.

4.1 Datasets

The datasets are detailed in Table 1. Our preprocessing steps are similar to [24]. We order the documents by timestamp to reflect the actual order of arrival in real time. (i) TREC: This dataset consists of 25868[4] tweets whch were judged relevant to 269 topics in the TREC microblog track[5]. (ii) uci_news: This is a subset[6] (10K articles) of the much larger (400K articles) UCI News Aggregator dataset in UCIML repository [11]. The "story" identifier gives the cluster label. We use only the titles of the news articles and ignore the content.

4.2 Baselines

We use the following models as baselines:

- GSDMM [25]. This is a Dirichlet Multinomial Mixture model for short text clustering without temporal dependency information.
- MStream and MStreamF [24]. MStream is an advanced algorithm for clustering short text streams based on the Dirichlet Process Multinomial Mixture Model [26]. MStreamF is a version of MStream which can forget outdated documents and do batch processing.
- OSDM [13]. This is a fully online model which improves on MStream(F) by removing the need for batch processing and including semantic information in the form of word-to-word co-occurrence matrix as a cluster feature.
- OSDMHP. This is a combination of OSDM with Hawkes Process similar to the combination of Dirichlet Process with the Hawkes Process in [10].
- DP-BMM [8]. This is a Dirichlet Process Biterm Mixture Model which considers biterms (word pairs) instead of words to address the sparsity of co-occurrence information in short documents.
- Rakib *et al.* [18]. This is a combination of several heuristic methods including using frequent word pairs and reassigning cluster outliers to more appropriate clusters using semantic information (word embeddings).
- DHP [10]. This is the classic Dirichlet-Hawkes Process without ICF prior.

For all algorithms we generally used the default parameter settings chosen by the authors. Since GSDMM requires the number of topics K to be set beforehand, we set $K = 300$ for TREC and $K = 500$ for uci_news. In addition for DP-BMM we set the hyper-parameter $\alpha = 1.5$. We denote DHP with ICF prior and its stacked version by DHP+ICF and SDHP+ICF respectively. For DHP as well as our methods we set $\lambda_0 = 0.1$ and use Sequential Monte Carlo sampling with 8 particles for inference. Results for DHP+ICF and SDHP+ICF are averaged over 10 trial runs.

4.3 Metrics

We use the following metrics to evaluate clustering results: Homogeneity (**Ho**), Completeness (**Co**), V-Measure (**VM**), Purity (**Pu**) and Normalized Mutual Information (**NMI**) (implemented with sklearn

[1]We have experimented without using the log weighting in this step, but results were slightly inferior.
[2]We empirically found that a third-order clustering yields very poor results, so we stop at second-order.
[3]We have experimented with using the log weighting in this step, but results were slightly inferior

[4]Originally 30322 tweets. We were able to collect less tweets due to suspension of some user accounts.
[5]http://trec.nist.gov/data/microblog.html
[6]https://www.kaggle.com/louislung/uci-news-aggregator-dataset-with-content

API[7]). **Ho** and **Pu** are high when each cluster has documents of a single topic. **Co** is high if each topic is represented by a single cluster. **VM** measures balance between **Ho** and **Co**. **NMI** measures overall quality of the clusters. **VM** and **NMI** are the most reliable metrics since **Ho** and **Pu** will be perfect if every document forms a singleton cluster, and **Co** will be perfect if all are in the same cluster.

4.4 Results

Algorithm	Ho	Co	Pu	VM	NMI
GSDMM	0.680	**0.821**	0.549	0.744	0.747
MStream	0.261	0.277	0.142	0.269	0.269
MStreamF	0.234	0.269	0.142	0.250	0.251
OSDM	0.628	0.446	0.401	0.522	0.530
OSDMHP	0.656	0.455	0.453	0.538	0.547
DP-BMM	0.831	*0.797*	0.755	0.814	0.814
Rakib *et al.* [18]	**1.0**	0.580	**1.0**	0.734	0.761
DHP	0.722	0.790	0.573	0.754	0.755
DHP+ICF	*0.924*	0.726	*0.865*	*0.818*	*0.819*
SDHP+ICF	0.894	0.763	0.827	**0.823**	**0.826**

Table 2: Experimental results on TREC dataset. The best and second-best figures are shown in bold and *italics* respectively.

Algorithm	Ho	Co	Pu	VM	NMI
GSDMM	0.701	*0.905*	0.445	0.790	0.796
MStream	0.662	0.774	0.344	0.713	0.716
MStreamF	0.739	0.816	0.451	0.776	0.777
OSDM	0.740	0.756	0.468	0.748	0.748
OSDMHP	0.830	0.826	0.601	0.828	0.828
DP-BMM	0.736	0.903	0.455	0.811	0.815
Rakib *et al.* [18]	**1.0**	0.739	**1.0**	0.850	0.860
DHP	0.773	**0.908**	0.522	0.835	0.838
DHP+ICF	*0.870*	0.890	*0.693*	**0.880**	**0.880**
SDHP+ICF	0.853	0.893	0.662	*0.872*	*0.872*

Table 3: Experimental results on uci_news dataset. The best and second-best figures are shown in bold and *italics* respectively.

Tables 2 and 3 show the experimental results on the TREC and uci_news datasets respectively. We see that for TREC, SDHP+ICF gives better NMI and V-Measure scores than DHP+ICF, but not for uci_news. The reason is that the average cluster temporal density is much higher for uci_news than TREC (see Table 1), so the Hawkes process describes the temporal dynamics of uci_news more accurately, making second order clustering redundant. This is also why OSDMHP performs significantly better than OSDM on uci_news, but only marginally better on TREC. Also, DHP performs much better on uci_news than on TREC. Among the discrete time baselines,

[7]http://scikit-learn.org

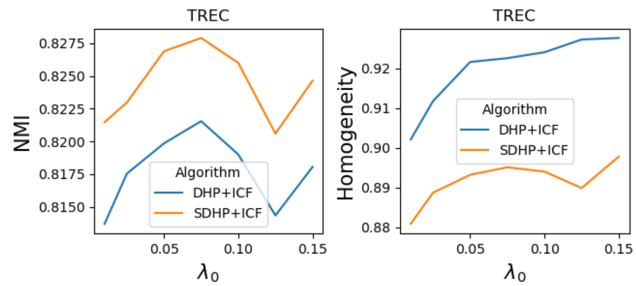


Figure 2: Variation of NMI and Ho vs λ_0 on TREC.

DP-BMM and [18] perform the best in terms of NMI and V-Measure scores, followed by GSDMM. Interestingly, OSDM, MStream and MStreamF perform significantly better on uci_news than on TREC, presumably because documents of the same ground truth topic are grouped very closely in uci_news due to high cluster temporal density.

We note that although DHP+ICF and SDHP+ICF are superior to baselines in terms of NMI and V-Measure scores, they are not better than the baselines in terms of Homogeneity, Completeness and Purity. This is mainly because these metrics are biased either towards small clusters (Homogeneity and Purity) or towards large clusters (Completeness). [18] uses an elaborate outlier removal mechanism to eliminate all but the most relevant documents, and hence forms small clusters. Hence it yields perfect Homogeneity and Purity scores, but fails to give good Completeness. Since GSDMM requires the number of clusters to be fixed beforehand, it typically forms large clusters and hence has a high Completeness score in both datasets. On the other hand, the Dirichlet Process based models such as DP-BMM and DHP are naturally biased towards large clusters (this can be said to be a natural propensity of the Dirichlet Process which employs the "preferential attachment" principle) and so perform well on the Completeness score. However, they lose out on Homogeneity and Purity. On the other hand, DHP+ICF and SDHP+ICF can strike a balance between Homogeneity and Completeness due to the ICF information (as well as the stacking in case of SDHP+ICF).

In Figure 2 we show the effect of varying λ_0 on TREC. We measure the performance in terms of NMI (for overall performance) as well as Homogeneity (to substantiate the claims made in §3.1). We see that in no case does DHP+ICF perform better than SDHP+ICF in terms of NMI. Furthermore, as discussed in §3.1, DHP+ICF always yields less homogeneity with decreasing λ_0 due to the second limitation of DHP which is addressed by SDHP+ICF.

5 CONCLUSION

In this paper we have proposed two modifications to the Dirichlet-Hawkes process described by [10], viz. using normalized ICF priors and a stacked version with a second order clustering. We experimentally show that these two approaches, though simple and intuitive, can perform at par with state-of-the art short text clustering methods. As future work, the DHP model may be improved by using biterms along the lines of [8] to address the problem of sparsity of word co-occurrence in short text.

REFERENCES

[1] Charu C Aggarwal. 2013. A Survey of Stream Clustering Algorithms.

[2] Amr Ahmed and Eric Xing. 2008. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *Proceedings of the 2008 SIAM International Conference on Data Mining*. SIAM, 219–230.

[3] Hesam Amoualian, Marianne Clausel, Eric Gaussier, and Massih-Reza Amini. 2016. 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*. 695–704.

[4] Peng Bao, Hua-Wei Shen, Xiaolong Jin, and Xue-Qi Cheng. 2015. Modeling and predicting popularity dynamics of microblogs using self-excited hawkes processes. In *Proceedings of the 24th International Conference on World Wide Web*. 9–10.

[5] David M Blei and John D Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*. 113–120.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[7] Junyang Chen, Zhiguo Gong, and Weiwen Liu. 2019. A nonparametric model for online topic discovery with word embeddings. *Information Sciences* 504 (2019), 32–47.

[8] Junyang Chen, Zhiguo Gong, and Weiwen Liu. 2020. A Dirichlet process biterm-based mixture model for short text stream clustering. *Applied Intelligence* (2020), 1–11.

[9] Wanying Ding, Yue Zhang, Chaomei Chen, and Xiaohua Hu. 2016. Semi-supervised Dirichlet-Hawkes process with applications of topic detection and tracking in Twitter. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 869–874.

[10] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. 2015. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 219–228.

[11] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[12] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[13] Jay Kumar, Junming Shao, Salah Uddin, and Wazir Ali. 2020. An Online Semantic-enhanced Dirichlet Model for Short Text Stream Clustering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 766–776.

[14] Shangsong Liang, Emine Yilmaz, and Evangelos Kanoulas. 2016. Dynamic clustering of streaming short documents. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 995–1004.

[15] Alireza Rezaei Mahdiraji. 2009. Clustering data stream: A survey of algorithms. *International Journal of Knowledge-based and Intelligent Engineering Systems* 13, 2 (2009), 39–44.

[16] Charalampos Mavroforakis, Isabel Valera, and Manuel Gomez Rodriguez. 2016. Modeling the dynamics of online learning activity. *arXiv preprint arXiv:1610.05775* (2016).

[17] Hai-Long Nguyen, Yew-Kwong Woon, and Wee-Keong Ng. 2015. A survey on data stream clustering and classification. *Knowledge and information systems* 45, 3 (2015), 535–569.

[18] Md Rashadul Hasan Rakib, Norbert Zeh, and Evangelos Milios. 2020. Short Text Stream Clustering via Frequent Word Pairs and Reassignment of Outliers to Clusters. In *Proceedings of the ACM Symposium on Document Engineering 2020*. 1–4.

[19] Yeon Seonwoo, Alice Oh, and Sungjoon Park. 2018. Hierarchical dirichlet gaussian marked hawkes process for narrative reconstruction in continuous time domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3316–3325.

[20] Jonathan A Silva, Elaine R Faria, Rodrigo C Barros, Eduardo R Hruschka, André CPLF de Carvalho, and João Gama. 2013. Data stream clustering: A survey. *ACM Computing Surveys (CSUR)* 46, 1 (2013), 1–31.

[21] Yu Wang, Eugene Agichtein, and Michele Benzi. 2012. TM-LDA: efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 123–131.

[22] Xing Wei, Jimeng Sun, and Xuerui Wang. 2007. Dynamic Mixture Models for Multiple Time-Series.. In *Ijcai*, Vol. 7. 2909–2914.

[23] Hongteng Xu and Hongyuan Zha. 2017. A Dirichlet Mixture Model of Hawkes Processes for Event Sequence Clustering. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/dd8eb9f23fbd362da0e3f4e70b878c16-Paper.pdf

[24] Jianhua Yin, Daren Chao, Zhongkun Liu, Wei Zhang, Xiaohui Yu, and Jianyong Wang. 2018. Model-based clustering of short text streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2634–2642.

[25] Jianhua Yin and Jianyong Wang. 2014. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 233–242.

[26] Jianhua Yin and Jianyong Wang. 2016. A model-based approach for text clustering with outlier detection. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 625–636.

[27] Jianhua Yin and Jianyong Wang. 2016. A text clustering algorithm using an online clustering scheme for initialization. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 1995–2004.

[28] Yukun Zhao, Shangsong Liang, Zhaochun Ren, Jun Ma, Emine Yilmaz, and Maarten de Rijke. 2016. Explainable user clustering in short text streams. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 155–164.