A Pitman-Yor Process Self-Aggregated Topic Model for Short Texts of Social Media

YUE NIU, HONGJIE ZHANG, AND JING LI
School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China
Corresponding author: Jing Li (lj@ustc.edu.cn)

ABSTRACT In recent years, with the rapid growth of social media, short texts have been very prevalent on the internet. Due to the limited length of each short text, word co-occurrence information in this type of documents is sparse. Conventional topic models based on word co-occurrence are unable to distill coherent topics on short texts. A state-of-the-art strategy is self-aggregated topic models which implicitly aggregate short texts into latent long documents. But these models have two problems. One problem is that the number of long documents should be defined explicitly and the inappropriate number leads to poor performance. Another problem is that latent long documents may bring non-semantic word co-occurrence which brings incoherent topics. In this article, we firstly apply the Chinese restaurant process to automatically generate the number of long documents according to the scale of short texts. Then to exclude non-semantic word co-occurrence, we propose a novel probabilistic model generating latent long documents in a more semantically way. Specifically, our model employs a pitman-yor process to aggregate short texts into long documents. This stochastic process can guarantee that the distribution between short texts and long documents following a power-law distribution which can be found in social media like Twitter. Finally, we compared our method with several state-of-the-art methods on four real short texts corpus. The experiment results show that our model performs superior to other methods with the metrics of topic coherence and text classification.

INDEX TERMS Text mining, short text, social media, topic model.
The self-aggregation strategy is another popular strategy. These methods automatically aggregate short texts into latent long documents and distill topics from these long documents without any auxiliary information. Any two words from different short texts constitute the new word co-occurrence. As each document is much longer, the word co-occurrence will be less sparse. SATM [19] is the first proposed self-aggregated model. However, this model will be overfitting when the size of a short text corpus grows. PTM [20] is another state-of-the-art model which can avoid overfitting. But all these models still have two problems.

One problem is that these methods need the number of long documents as the explicit input. This attribute will strongly influence the performance of the model. The inappropriate number of long documents will lead to poor performance. SPTM [20] applies Spike and Slab prior to improve the poor performance caused by the small number of long documents. But this model can only provide limited improvement and still suffers the problem of the inappropriate long document number.

Another problem is that latent long documents will bring non-semantic word co-occurrence. Ideally, the content of short texts aggregated in one long document is semantically related. So the word co-occurrence incorporated is also semantically related. But all these self-aggregated models cannot guarantee that short texts are aggregated as expected. Non-semantic related short texts are likely to be aggregated in one long document and bring plenty of non-semantic word co-occurrence. And non-semantic word co-occurrence will lead to incoherent topics.

Firstly, ascertaining an appropriate number of long documents is not easy work. Because this number should not be too small or too large compared with the number of short texts. If we change the number of short texts, the appropriate number of long documents should also be changed. Otherwise, the number of long documents may be too small or too large. Long documents with a small number will incorporate plenty of non-semantic word co-occurrence and a large number will lack additional word co-occurrence. So if we can expand or reduce the number of long documents according to the number of short texts, we can make up for this deficiency. Inspired by this, we apply the Chinese restaurant process [21] which constructs an exchangeable sequence of short texts and each short text will sample an existing long document or creates a new long document in a certain probability. So this process can sample the number of long documents. By defining the prior of long documents’ dispersion, this process can automatically generate the appropriate number no matter how the number of short texts changes.

Secondly, after investigating short texts on social media, we found an interesting phenomenon. We collected a corpus of tweets and each tweet has a label. Then we calculated the number of tweets of each hashtag. Figure 1 shows that if we take the logarithm of labels’ serial numbers and the corresponding number of tweets, we can find an approximately linear function. This observation is called Zipf’s law and proves that the relation of tweets and labels follows the power-law distribution. Of course, tweets with the same label are semantically related.

This phenomenon inspired us that semantically related short texts may follow a power-law distribution. So if we aggregate short texts following such distribution, short texts in one long document are more likely to be semantically related. We adopted a pitman-yor process to generate a power-law distribution between short texts and long documents. Then as word co-occurrences are not sparse in these long documents, we generate topics from these documents.

The contributions of our pitman-yor process self-aggregated topic model (PYSTM) are as follows:

- By sampling from the Chinese restaurant method, PYSTM will generate the number of long documents. Our model can avoid users inputting this number explicitly and automatically generate the appropriate number following the scale of corpus to improve performance.
- We propose the pitman-yor process to generate long documents. This pitman-yor process can capture the power-law phenomenon of short texts and generate semantically related short texts into one long document. Therefore, less noisy information is brought to the result.
- We compare our model with several state-of-the-art methods on several real-world short texts corpus. The experimental results demonstrate our model’s superiority in terms of topic coherence and classification accuracy.

The rest of the paper is organized as follows. In Section 2, we discuss related works. In Section 3, we propose the generation process of our model PYSTM and give the inference details. Then we present the experimental results in Section 4 and finally conclude our work in Section 5.

II. RELATED WORK

One strategy is incorporating meta-data into topic models. Aggregating short texts according to the meta-data of the corpus is a straight forward way. Some models [11], [22] aggregate short texts generated by the same author. ET-LDA [23] aggregates tweets associated with the same event. Pooling [12] method aggregates tweets with the same
labels. Then frameworks [13], [24] combining different types of context information are generated. mLDA [13] combines authors and labels and rrPLSA [24] combines authors and social roles. But for all these models, meta-data information is not always available.

Other state-of-the-art models incorporating word embeddings [25], [26]. Models train word embeddings from an auxiliary document set and incorporate this information as the compensation information. LF-LDA and LF-DMM [27] are the earliest models incorporating word embeddings. They use word embeddings to estimate the probability of words. Gaussian LDA [28] treats each document as a collection of word embeddings. GPU-DMM [29] uses Polya urn [30] model generating topics according to word embeddings. A global and local model GLTM [16] integrates both word embeddings trained from short texts corpus and auxiliary corpus. TRNMF [17] uses word embeddings to generate sentence similarity regularization and integrates with word co-occurrence. CME-DMM [31] is a collaboratively modeling and embedding framework incorporating topic and word embeddings. WE-PTM [32] uses word embeddings to generate the prior of topic-word distribution instead of defining a symmetric prior. However, the performance of all embedding methods relies on an appropriate auxiliary corpus. If the auxiliary dataset mismatches short texts, the performance will be very poor [18].

Another strategy is proposing a customized model for short texts without auxiliary information.

There are some other strategies. DMM methods [7], [33] and Dual-Sparse Model [9] suggest that each text is sampled from only one or a few topics. But this suggestion is only suitable for some corpus. BTM [34] and WNTM [35] use global word co-occurrence which aggregates all word co-occurrence in the corpus. But these two models lack direct representation for documents.

SATM [19] is the first method aggregates short texts into latent long documents according to latent topics. But this model has a problem of overfitting and is time-consuming. PTM [20] proposes a simple but effective generative process that incorporates a latent long document variable layer between short texts and topics. SPTM [20] applies Spike and Slab prior to improve the poor performance when a small number of long documents is defined. But all existing self-aggregated topic models have two problems. Firstly, the number of long documents may be inappropriate which will lead to poor performance. Secondly, they cannot avoid incorporating non-semantic word co-occurrence.

III. MODEL AND INFERENCE

In this section, we propose our Pitman-yor Process Self-aggregated Topic Model(PYSTM) customized for short texts. Firstly, we introduce the generative process and the graphical model representation of PYSTM. Then, we show how this model can automatically sample the number of long documents and how it can follow the power-law distribution.

Finally, we exhibit the inference method according to the Gibbs sampling scheme [36].

A. OVERVIEW

Traditional topic models like LDA designs a generation process. These models suggest that one document is generated following a process as follows: For each word $w$ in document $i$, firstly sample a topic $z$ from $i$ according to a multi-nominal distribution $P(z|i)$. Then sample the word $w$ from topic $z$ according to another multi-nominal distribution $P(w|z)$. Then a joint probability distribution is generated as $P(i, z, w) = P(w|z)P(z|i)$. In this distribution, $i$ is the variable of the document, $w$ is the variable of the word and $z$ is the latent variable topic. Because $i$ and $w$ are already known, so we only need to sample $z$ according to $P(i, z, w)$ by Gibbs sampling method.

But in self-aggregated models like PTM, the model incorporates another latent variable: latent pseudo documents $l$. The generation process is as follows: For each short text, latent long document $l$ is generated from short text $i$ according to a multi-nominal distribution $P(l|i)$. Then, latent topic $z$ is generated from long document $l$ according to another multi-nominal distribution $P(z|l)$. Finally, each word $w$ in short text $i$ is generated from topic $z$ according to another multi-nominal distribution $P(w|z)$. So the joint probability distribution is $P(i, l, z, w) = P(w|z)P(z|l)P(l|i)$. Two latent variables $z$ and $l$ will be sampled according to this distribution. If the number of long documents is restricted, defining this number less than short texts and more than topics, these documents will be documents aggregated by short texts.

In model PTM, distributions $P(w|z)$, $P(z|l)$, and $P(l|i)$ are all multi-nominal distributions. Especially for $P(l|i)$, short texts share one multi-nominal distribution to generate long documents. Distribution $P(l|i)$ can be seen as a dice, sides of this dice are long documents. Each short text will roll this dice to sample a long document. The probability of the upside is only related to samples of topics. So, this sampling procedure cannot avoid aggregating semantic relatedness short texts into one long document.

To overcome this problem, we use the Chinese restaurant process and pitman-yor process instead of multi-nominal distribution in $P(l|i)$. That means we generate long documents as follows: Firstly, we generate the number of long documents according to the Chinese restaurant process. Then along with this procedure, we sample the probability of short text $i$ selecting long document $l$ not only according to topics but also according to the power-law distribution by the pitman-yor process.

Because the Chinese restaurant process and the pitman-yor process are specific sampling methods of the Dirichlet process (DP) [21]. So we use DP as a generalized description in the generation process of our model. Dirichlet process is a stochastic process that can sample a probability measure over a subset of one space. For the relationship between long documents and short text, long documents can be seen as
sampling a distribution and choosing a dice from it means choosing a dice from a dice set. There are different dices with different numbers of sides. Choosing a dice from a dice set can be viewed as choosing a dice from a dice set with different probabilities. This process can be viewed as choosing a dice to sample long documents. The details of the generation process of PYSTM can be referred to as follows.

1) Sample $G \sim DP(\alpha, H)$
2) For each topic $z$
   - Sample $\eta_z \sim Dir(\gamma)$
3) For each latent long document $l$
   - Sample $\theta_l \sim Dir(\beta)$
4) For each short document $i$
   - Sample a latent long document $l \sim G$
   - For each word $w_{i,l}$ in short document $i$
     - Sample a topic $z \sim Multi(\theta_l)$
     - Sample the word $w_{i,l} \sim Multi(\eta_z)$

This generation process is the generalized mathematical representation of our model. Firstly, we sampled a distribution $G$ from $DP(\alpha, H)$. Secondly, for each short text $i$, we sample long document $l$ according to $G$. Then, for each word $w$ in $i$, we sample a topic $z$ from $l$ according to a multinomial distribution $P(z|l)$. Finally, we sample word $w$ from topic $z$ according to a multinomial distribution $P(w|z)$. So the joint probability distribution of our model is $P(i, l, z, w) = P(w|z)P(z|l)G$.

Here $G$ is a discrete probability distribution with a finite size. In our model, we define how $G$ is sampled from $DP(\alpha, H)$. We use the Chinese restaurant process to sample $G$ along with a sampled size according to the scale of short texts. Also in this procedure, we use the pitman-yor process to make sure $G$ is a sample following the power-law distribution. In this way, by defining the Dirichlet process, we define how short texts are generated into long documents. The following two sections will describe details about the Chinese restaurant process and the pitman-yor process.

The graphical model can be seen in Figure 2. The required notations in this paper are shown in Table 1.

**TABLE 1. Definitions of notations.**

| Notation | Meaning |
|----------|---------|
| $T$      | number of topics |
| $V$      | number of vocabulary |
| $N$      | number of short texts |
| $N_w$    | number of words of each short text |
| $G$      | probability measure of Dirichlet process |
| $L$      | latent variables of long documents |
| $Z$      | latent topics |
| $W$      | word in short texts |
| $\alpha$ | concentration parameter of Dirichlet process |
| $\beta$  | base measure of Dirichlet process |
| $\gamma$ | beta prior for $\beta$ |
| $\theta$ | multinomial distribution over topics of long documents |
| $\eta$   | multinomial distribution over words of topics |
| $DP(\cdot)$ | Dirichlet process |
| $Dir(\cdot)$ | Dirichlet distribution |
| $Multi(\cdot)$ | Multinomial distribution |
| $PY(\cdot)$ | Pitman-yor process |

**B. CHINESE RESTAURANT PROCESS**

To automatically generate the number of long documents, we adopt the Chinese restaurant process to describe the distribution over partitions drawing from the Dirichlet process. This method is described as follows:

1) suggest that a Chinese restaurant has unlimited number of tables
2) the first customer sits at the first table
3) the $n$th customer sits at
   a) the table $k$ with probability $\frac{n_k}{\alpha_0+n-1}$
   b) a new table with probability $\frac{\alpha}{\alpha_0+n-1}$

Here $n_k$ means the number of customers already sits at table $k$. And $\alpha_0$ is the concentration parameter. In our model PYSTM, customers are short texts and tables are long documents. Therefore, for each short text, there are two conditions for sampling a long document. One is sampling a document from existing documents. Another is creating a new document and sampling it. Repeating this sampling procedure for each short text, we automatically generate a long document set. Finally, we can get the probability $\phi_i$ of sampling $k$th long document for $i$th short text:

$$
\phi_i|\phi_1, \ldots, \phi_{i-1}, \alpha, H \sim \sum_{k=1}^{K} \frac{n_k}{\alpha+n-1} \delta_{\theta_k} + \frac{\alpha}{\alpha+n-1} H
$$

In this probability equation, $n$ means the number of short texts, $K$ means the number of long documents, $n_k$ means the number of short texts of the $k$th long document, and $\theta_k$ corresponds to the $k$th long document. The probability will be $\frac{n_k}{\alpha+n-1}$ if the short text is aggregated to the $k$th long document. And a new long document will be created with the probability of $\frac{\alpha}{\alpha+n-1}$. From this equation, we can find that the Chinese restaurant process provides a probability distribution about number $K$, and we can sampling $K$ according to $n$ short texts. The expected value of $K$ is $\sum_{i=1}^{n} \frac{\alpha}{\alpha+i-1}$ where $i$ means the $i$th short text. So with a constant value of $\alpha$, the number of long documents will change depending on the scale of short texts.

**C. PITMAN-YOR PROCESS**

In order to avoid incorporating too many non-semantic word co-occurrence, we need to sample long documents in a more
reasonably. In real scenes, we find that the semantic aggregation of short texts follows a power-law distribution. Therefore, we expect the distribution of the long documents generation process also follows the power-law distribution. As we sample long documents according to the Chinese restaurant process, we employ pitman-yor process to determine the expected distribution. The pitman-yor process is also a Dirichlet process that has two parameters. We write \( G \sim PY(d, \alpha, H) \) as the formal description. The value of prior \( d \) should be \( 0 < d < 1 \). The marginalized distribution is as follows:

\[
\phi_i | \phi_1, \ldots, \phi_{i-1}, \alpha, H \sim \sum_{k=1}^{K} \frac{n_k - d}{\alpha + n - 1} \delta_{\phi_k} + \frac{\alpha + Kd}{\alpha + n - 1} H
\]

(2)

According to Equation 2, we can find two salient properties showing how pitman-yor process yields power-law behavior.

The first property is rich-get-richer. If a short text \( i \) samples a long document \( k \) that already exists, the probability will be \( \frac{n_k - d}{\alpha + n - 1} \). Then document \( k \) becomes larger. The new value of \( n_k \) will be \( n_k + 1 \). For the following short text \( i^* \), it will sample document \( k \) with a higher probability compared to \( i \). If \( k \) is sampled by \( i^* \), the value of \( n_k \) will grow larger. Therefore, when a long document \( k \) is bigger than others, it will have a higher probability to be sampled. A higher probability will then make \( k \) grow bigger. Finally, after sampling all long documents, there exists a few long documents that are much bigger than others as a big aggregation of short texts.

The second property is that there are a large number of long documents sampled by a small set of short texts. The probability of sampling a new long document \( k \) + 1 by short text \( i \) is \( \frac{\alpha + Kd}{\alpha + n - 1} \). We can find that when the number of long documents \( K \) grows bigger, the probability of sampling a new document will be higher. Along with the short texts sampling process, more and more new long documents are generated containing only a few short texts. But these newly created documents only have a small \( \phi_k \), which means short texts will sample this document with a low probability. Hence, when the process is finished, documents with small \( \phi_k \) are more likely to remain small. Combining the first property, the length and the number of long documents sampled will follow a typical power-law distribution.

D. Inference

In PYSTM, the exact posterior inference is intractable. So we train our model with collapsed Gibbs sampling method. By integrating out some parameters, we sample two latent variables: long document \( l \) and topic \( z \).

1) SAMPLING LONG DOCUMENTS ASSIGNMENTS \( l \)

For a short text \( i \), the probability function of sampling long document \( l \) is designed as follows:

\[
p(l_i = l | l_{-i}, w, z, \alpha, d, \beta, \gamma) = \frac{p(l_i | \alpha, d)}{p(l_{-i} | \alpha, d)} \cdot \frac{p(w, z | l_i, \beta, \gamma)}{p(w, z | l_{-i}, \beta, \gamma)}
\]

(3)

\( l_i \) means the sampled long document by short text \( i \). And \( l_{-i} \) means long documents excluding \( l \) sampled by short text \( i \).

According to Equation 3, we can divide this assignment into two parts.

Firstly, we generate the probability of \( \frac{p(l_i | \alpha, d)}{p(l_{-i} | \alpha, d)} \). That is,

\[
p(l_i | \alpha, d) \propto \begin{cases} n_k - d, & k < K \\ \alpha + Kd, & k = K + 1 \end{cases}
\]

(4)

Because the sequence of the Chinese restaurant process is exchangeable, the short text \( i \) can be regarded as the last one of the sequence. Sampling \( l \) can be divided into two situations. Condition on \( k < K \), means \( l \) is sampled from long documents that already exist. Condition on \( k = K + 1 \), means we create a new long document as the sample.

Secondly, we generate the probability of \( \frac{p(w, z | l_i, \beta, \gamma)}{p(w, z | l_{-i}, \beta, \gamma)} \). That is,

\[
p(w, l_i | \beta, \gamma) \propto \begin{cases} p(z_{-i} | l_i, \beta) p(w | z_{-i}, \beta, \gamma) \\ p(z_{-i} | l_{-i}, \beta) p(w | z_{-i}, \beta, \gamma) \end{cases}
\]

(5)

where \( z_{-i} \) means topics excluding short text \( i \). If long document \( l \) is sampled from the existing long documents, by integrating out \( \theta \), we obtain the probability as follows:

\[
p(z_{-i} | l_i, \beta) \propto \begin{cases} \prod_{t=1}^{T} \sum_{i=1}^{n_t} (n_t + \beta + j - 1) \\ \prod_{t=1}^{T} (n_t + T \beta + g - 1) \end{cases}
\]

(6)

In this equation, \( n_t \) means the number of topic \( t \) in long document \( l \) excluding counts from \( i \). \( n_t \) means the number of topic \( t \) in short text \( i \). \( t \) means each topic occurred in short text \( i \). \( \beta \) means the length of short text \( i \). \( T \) means the number of short texts. And \( n_t \) means the length of short text \( i \).

If long document \( l \) is sampled from a new document, we obtain the probability as follows:

\[
p(z_{-i} | l_i, \beta) \propto \begin{cases} \prod_{t=1}^{T} \sum_{i=1}^{n_t} (n_t + \beta + j - 1) \\ \prod_{t=1}^{T} (n_t + T \beta + g - 1) \end{cases}
\]

(7)

Finally, we obtain the sampling probability by integrating the two parts. That is,

\[
p(l_i = l | l_{-i}, w, z, \alpha, d, \beta, \gamma) \propto \begin{cases} \prod_{t=1}^{T} \sum_{i=1}^{n_t} (n_t + \beta - j) \\ \prod_{t=1}^{T} (n_t + T \beta - g) \end{cases}, & k < K \\ (\alpha + Kd) \begin{cases} \prod_{t=1}^{T} \sum_{i=1}^{n_t} (n_t + \beta + j - 1) \\ \prod_{t=1}^{T} (n_t + T \beta + g - 1) \end{cases}, & k = K + 1
\]

(8)

2) SAMPLING TOPICS ASSIGNMENTS \( z \)

For a long document \( l \), the approach of sampling the topic \( z \) of word \( w \) is designed as follows:

\[
p(z_t = z | z_{-w}, l, \beta, \gamma) \propto \begin{cases} p(z_t | l, \beta) p(w | z_{-w}, \gamma) \\ p(z_t | l, \beta) p(w | z_{-w}, \gamma) \end{cases}
\]

(9)

Here, \( z_{-w} \) means set of topics excluding topic \( z \) of word \( w \).
Firstly, we obtain the probability of \( \frac{p(z|l, \beta)}{p(z_{-w}|l, \beta)} \) by integrating out \( \theta \). That is,

\[
p(z|l, \beta) \propto n_{i,z} + \beta - 1
\]

(10)

\( n_{i,z} \) means the number of topic \( z \) in long document \( l \). Then, we obtain the probability of \( \frac{p(w|z, \gamma)}{p(w|z_{-w}, \gamma)} \) by integrating out \( \eta \). That is,

\[
p(w|z, \gamma) \propto \frac{n_{w,z}^{w} + \gamma - 1}{n_{w,z}^{w} + V \gamma - 1}
\]

(11)

In this equation, \( n_{w,z}^{w} \) means the number of words the same word \( w \) of topic \( z \) excluding topic \( w \) of word \( w \). \( n_{z}^{w} \) means the number of total words of topic \( z \) excluding word \( w \). \( V \) means length of vocabulary. Finally, we obtain the sampling probability. That is,

\[
p(z = z_{-l}, w, l, \beta, \gamma) \propto (n_{l,t} + \beta - 1) \frac{n_{w,z}^{w} + \gamma - 1}{n_{w,z}^{w} + V \gamma - 1}
\]

(12)

According to the equations above, the detailed sampling process of our model is illustrated in Algorithm 1.

**Algorithm 1** Gibbs Sampling of PYSTM

**Input:** parameters \( \alpha, \beta, \gamma, \eta \)

**Output:** topic-word distribution

1: initialize all variables

2: while iteration < Maxiteration do

3: for \( i \in \{1, N\} \) do

4: \( l = l_{i} \);

5: \( n_{k} = n_{k} - 1 \);

6: \( n_{l,t} = n_{l,t} - n_{l,i} \);

7: \( n_{l} = n_{l} - n_{l} \);

8: Sample long document \( l_{i} \) according to Equation 8;

9: if \( l_{i} \) is a new long document then

10: \( K = K + 1 \);

11: end if

12: \( l = l_{i} \);

13: \( n_{k} = n_{k} + 1 \);

14: \( n_{l,t} = n_{l,t} + n_{l,i} \);

15: \( n_{l} = n_{l} + n_{l} \);

16: for \( w \in \{1, N_{w}\} \) do

17: \( z = z_{w} \);

18: \( n_{l,z} = n_{l,z} - 1 \);

19: \( n_{w,z} = n_{w,z} - 1 \);

20: \( n_{z} = n_{z} - 1 \);

21: Sample topic \( z_{w} \) according to Equation 12;

22: \( z = z_{w} \);

23: \( n_{l,z} = n_{l,z} + 1 \);

24: \( n_{w,z} = n_{w,z} + 1 \);

25: \( n_{z} = n_{z} + 1 \);

26: end for

27: end for

28: end while

### IV. EXPERIMENTAL RESULTS

In this section, firstly we introduce the datasets in our experiment, the methods for comparison, and two evaluation measures. Then we show results compared to the state-of-the-art models.

#### A. EXPERIMENTAL SETUP

1) **DATASETS**

In social media, we can find different kinds of short texts like news titles, microblogs, search result snippets, and descriptions of pictures. To represent these kinds of short texts, we adopt four datasets: News, Tweets, Snippets, and Captions. The information summary of these datasets is listed in Table 2, where \( D \) means the number of short texts, \( V \) indicates the size of the vocabulary, \( Len \) represents the average length of each short text and \( C \) shows the number of clusters of a corpus.

We minimally pre-processed these datasets by removing stopwords and words that occur only once. The brief description is as follows:

**News:** This dataset of news titles is collected from RSS feeds of three popular newspaper websites (nyt.com, usatoday.com, reuters.com). We generate 10,000 news titles across 7 categories (Sport, Business, U.S., Health, Sci&Tech, World and Entertainment).

**Tweets:** This dataset of tweets is collected from microblogs with hashtags. Each tweet with the same hashtag is considered to belong to the same category. It includes 100 categories and a total of 2,500 tweets.

**Snippets:** This dataset is collected from the results of web search transactions [37]. We generate 10,000 short texts and classifying texts into 8 domains: Business, Computers, Culture Arts, Education-Science, Engineering, Health, Politics-Society, and Sports, respectively.

**Captions:** This dataset contains 4,834 captions solicited from Mechanical Turkers for photographs from Flickr and Pascal [38]. Captions are divided into 20 categories.

2) **METHODS**

In this section, we introduce some state-of-the-art methods implemented for comparison.

a: **LDA**

LDA model is one of the most classical topic models. We implement this method as the base method.

b: **DSTM**

This method allows each document to select only a few focused topics and each topic to select its focused terms by using Spike and Slab prior.

### Table 2. Statistics of data sets.

| DataSet  | \( D \)  | \( V \)  | \( Len \) | \( C \) |
|----------|----------|----------|----------|-------|
| News     | 10,000   | 10,906   | 12       | 7     |
| Tweets   | 25,000   | 4,981    | 8        | 100   |
| Snippets | 10,000   | 4,705    | 14       | 8     |
| Captions | 4,834    | 3,165    | 5        | 20    |
c: SATM
This method is the first proposed self-aggregated topic model that follows two phases. The first phase follows the assumption of standard topic models to generate a set of regular-sized documents. The second phase generates a few short text snippets from these documents.

d: PTM
This method also proposes a self-aggregated model which incorporates latent long documents. By generating long documents from short texts and generating topics from long documents, this process will incorporate plenty of additional word co-occurrence.

e: SPTM
This method incorporates Spike and Slab prior to improve performance when the number of long documents is defined as too small. It suggests that each long document is more likely to generate a few topics.

3) PARAMETER SETTINGS
For Gibbs sampling process of all methods, we perform 4,000 total iterations and ignore first 1,000 samples to skip the convergence phase. For LDA, we use the common settings that $\alpha = 0.1$ and $\beta = 0.01$ according to paper [27]. For DSTM, we set $\pi = 0.1$ and $\gamma = 0.01$ according to paper [20] which finds that these numbers outperform settings suggested by paper [9]. For SATM, we set the number of long documents as 300, $\alpha = 50/T$ and $\beta = 0.1$ following settings suggested by paper [19]. For PTM, we set the number of long documents as 1000, $\alpha = 0.1$ and $\beta = 0.01$. For SPTM, we set the number of long documents as 1000, $\alpha = 0.1$, $\beta = 0.01$, $\gamma_0 = 0.1$ and $\bar{a} = 10^{-12}$. Settings of PTM and SPTM are all suggested settings in paper [20]. For our model PYSTM, we set $\alpha = 1000$, $d = 0.8$, $\beta = 0.1$ and $\gamma = 0.01$. The code for PYSTM is available at:\footnote{https://github.com/overlook2021/PYSTM.git}

4) EVALUATION MEASURES
In this section, we introduce two evaluation measures to evaluate the quality of topics and taxonomy relations.

a: TOPIC COHERENCE
We calculate the point-wise mutual information (PMI) [39] of each word pair to measure the coherence of topics. Computing PMI score needs an appropriate auxiliary corpus. So we choose the latest dump of Wikipedia articles as the external corpus which contains 5 million documents and 14 million words of vocabulary. We build a sliding window of 10 words and compute the PMI score as the following equation.

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

\[13\]

$w_i$ and $w_j$ is a word pair that occurred in one sliding window, and $p(w_i)$ is the marginal probability of word $w_i$ appearing in the sliding window. Then we compute the joint distribution $PMI(w_i, w_j)$ of word pairs in Wikipedia. Finally, we calculate the average PMI score of word pairs for top-N words of a given topic.

b: CLASSIFICATION ACCURACY
We conduct an external task of text classification to evaluate topic models. For each short text, topics are regarded as features, and probabilities from topic-document distributions are regarded as features’ values. Then we calculate a Support Vector Machine [40] classifier and compute the accuracy of classifications through bootstrap [41] method on short texts. Finally, we obtain the average accuracy of classifications as the measure.

B. TOPIC EVALUATION BY TOPIC COHERENCE
We calculate the PMI score to evaluate topic coherence. We generate 5, 10 and 15 topics and choose the top-10 words for each topic base on the probability. The results are shown in Figure 3.

From the results, we can observe that our method PYSTM outperforms other methods on all datasets with different numbers of topics. Compared to the base method LDA, the superior performance of our method demonstrates that aggregating short texts following the power-law distribution can overcome the sparsity of word co-occurrence. DSTM provides a little improvement compared to LDA and the performance may be worse in some cases. This poor performance shows that although DSTM generates a few focused topics of each short text, this method still lacking word co-occurrence. SATM performs the worst on four datasets. Possibly, SATM has an overfitting problem and needs a large dataset of training. But SATM is also time-consuming, which restricts the ability to compute large datasets. PTM and SPTM have similar performances. It is because the number of long documents is both set to be 1000. SPTM cannot improve the performance when the scale of long documents is a normal size like 1000. For PYSTM, the power-law distribution brings more semantic word co-occurrence. Therefore, our method performs superior compared to PTM and SPTM which generates more coherent topics.

C. TOPIC EVALUATION BY SEMANTICS
Topics constructed by words should also be readable by human beings. To analyze whether topics of our model are semantically meaningful, we train PYSTM on dataset News to generate top-10 words of 10 topics. Results can be seen in Table 3.

By analyzing words in these topics, we can interpret topics into semantic meanings. The concept of Topic 1 and Topic 5 are political news. Topic 1 is about energy and Topic 5 is about terrorists. Topic 2 is about economic news of some companies. Topic 3 is legal news. Topic 4, Topic 7, Topic 8, and Topic 9 are all sports news. Topic 4 is about basketball, Topic 7 is about the horse race, Topic 8 is about baseball and Topic 9 is about tennis. Topic 6 is entertainment news about
music. Topic 10 is health news. Results show that all semantic meanings of these topics can be easily understood by humans.

**D. TOPIC EVALUATION BY CLASSIFICATION**

We compare all models by performing document classification tasks. We generate 5, 10, and 15 topics as features of each short text and topic-document distributions as features’ value. Then we use the bootstrap method to sample training sets and testing sets. After 1,000 iterations of sampling, the accuracy results are shown in Figure 4.

These results show that our method PYSTM obtains the best performances on all datasets with different numbers of topics. As the base method, LDA outperforms DSTM on datasets of News, Tweets, and Snippets. Possibly, DSTM needs inferring a large number of priors that lead to poor performance. SATM obtains the worst performance, which is similar to the result of topic coherence. The reason is still the overfitting problem. Especially, the scale of data of each category becomes smaller compared with the overall scale of the dataset. PTM and SPTM have almost the same accuracies and obtain second best or third best results. But in the dataset Snippets, PTM and SPTM obtain poor performances compared to LDA. It may be because the distributions of topic-document of PTM and SPTM are generated indirectly.

As shown in equation

$$p(z|d) = p(z|l)p(l|d),$$

(14)

where $z$ means topics, $d$ means short texts and $l$ means long documents. Non-semantic word co-occurrence in $l$ will bring a lot of noisy information to $p(t|d)$. And this noisy...
information leads to poor performances when classifying short texts according to $p(t|d)$. Topic-document distribution of our model PYSTM is also similar to equation 14. But PYSTM can generate more semantic word co-occurrence information into long documents by incorporating the power-law distribution. Therefore, on dataset Snippets, PYSTM performs superior compared to LDA.

E. TOPIC EVALUATION BY CHANGING THE SCALE OF DATASET

The state-of-art model PTM needs explicitly to define the number of long documents. When the scale of the dataset changes, the appropriate number of long documents should also be changed. But our model PYSTM can automatically change the appropriate number according to the scale of the dataset. To prove this, we design an experiment on 4 datasets of News with different scales varied from 2000 to 10,000. Then we calculate the PMI score for PTM and PYSTM with different parameters. For PTM, the number of long documents is varied from 500 to 1000. For PYSTM, prior $\alpha$ controls the scale of long documents. So we vary $\alpha$ from 1000 to 2000. If the value of $\alpha$ is larger, the number of long documents will be sampled larger. Results are shown in Figure 5.

By changing the scale of News, we can find two different performances according to the number of long documents. For PTM, on dataset contains 2000 and 3000 news titles, the parameter of 500 long documents performs superior compared to 800 or 1000 long documents. Results show that the numbers 800 and 1000 are not appropriate. But when the dataset grows larger, the parameter of 800 long documents performs the best. And if the dataset grows more than one order of magnitude, the appropriate number changes to 1000. These results illustrate two facts. The first fact is the different number of documents leads to different performances. And the inappropriate number may result in very poor performance. The second one is when the scale of the dataset grows, the appropriate number also grows. Especially, if the scale of the dataset changes more than a certain range, the appropriate number must be changed. For PYSTM, the results are quite different. Firstly, models with each value of $\alpha$ outperform PTM method on all datasets. $\alpha = 1k$ obtains superior performance no matter how the dataset grows. Exactly, ordering $\alpha$ according to their performances, we find that the sequence is never changed on 4 datasets. Therefore, results show that PYSTM can retain the number of long documents to be appropriate with a fixed value of $\alpha$. 
F. EFFICIENCY ANALYSIS

We evaluate the efficiency of these models by calculating the average time of one iteration. All models are implemented by java on dataset Snippets and are computed on the same hardware environment. Results about initialization time and average time of each iteration are shown in Table 4. The unit of time is milliseconds.

As shown in the results, LDA is the most efficient model and also the simplest model. DSTM is the most inefficient model. It has many variables to infer and of course being time-consuming. SATM is the second inefficient model. It needs calculating $K$ long documents of $P$ conditions which also leads to time-consuming. Our model PYSTM is slightly slower compared to PTM due to an additional sampling procedure for the number of long documents. And, PYSTM is more efficient compared to SPTM. SPTM needs sampling topic selector variables which consumes more time. Finally, the time complexity of our model is $O(K + T)$. Here $K$ is the total number of long document variables for sampling. $T$ is the total number of topic variables for sampling.

V. CONCLUSION

In this paper, we propose a pitman-yor process self-aggregated topic model (PYSTM) for short texts. State-of-the-art self-aggregated topic models need explicitly define the number of long documents. By incorporating the Chinese restaurant process, our model can automatically generate the number of long documents according to the scale of datasets. Then previous self-aggregate topic models aggregate short texts following a multi-nominal distribution. But short texts without semantic relations may be generated into one long document. We find that the semantic relations of short texts in social media follow a power-law distribution. So we use the pitman-yor process to aggregate short texts according to this distribution. Then short texts that are semantically related are more likely to be generated into one long document and finally decrease the number of non-semantic word co-occurrence. Extensive experiments are conducted on 4 real-world datasets. Experiment results show that our model outperforms state-of-the-art methods. Currently, this work is focused on social media. In the future, we will study how to propose a more general model for different domains of short texts. And we will attempt to incorporate other semantic information such as correlative contextual semantic information.

REFERENCES

[1] D. M. Blei, “Probabilistic topic models,” Commun. ACM, vol. 55, no. 4, pp. 77–84, Apr. 2012.
[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” J. Mach. Learn. Res., vol. 3, pp. 993–1022, Mar. 2003.
[3] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, “Hierarchical Dirichlet processes,” J. Amer. Statist. Assoc., vol. 101, no. 476, pp. 1566–1581, Dec. 2006.
[4] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and A. X. Li, “Comparing Twitter and traditional media using topic models,” in Proc. Eur. Conf. Inf. Retr. Springer, 2011, pp. 338–349.
[5] X. Wang and A. McCallum, “Topics over time: A non-Markov continuous-time model of topical trends,” in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2006, pp. 424–433.
[6] S. Likhitha, B. S. Harish, and H. Kumar, “A detailed survey on topic modeling for document and short text data,” Int. J. Comput. Appl., vol. 178, no. 39, pp. 1–9, Aug. 2019.
[7] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, “Text classification from labeled and unlabeled documents using EM,” Mach. Learn., vol. 39, nos. 2–3, pp. 103–134, 2000.
[8] G. Yu, R. Huang, and Z. Wang, “Document clustering via Dirichlet process mixture model with feature selection,” in Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2010, pp. 763–772.
[9] T. Lin, W. Tian, Q. Mei, and H. Cheng, “The dual-sparse topic model: Mining focused topics and focused terms in short text,” in Proc. 23rd Int. Conf. World Wide Web, 2014, pp. 539–550.
[10] H. Ishwaran and J. S. Rao, “ Spike and slab variable selection: Frequentist and Bayesian strategies,” Ann. Statist., vol. 33, no. 2, pp. 730–773, 2005.
[11] L. Hong and B. D. Davison, “Empirical study of topic modeling in Twitter,” in Proc. 1st Workshop Social Media Anal., 2010, pp. 80–88.
[12] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie, “Improving LDA topic models for microblogs via tweet pooling and automatic labeling,” in Proc. 36th Int. Conf. Res. Develop. Inf. Retr., 2013, pp. 889–892.
[13] J. Tang, M. Zhang, and Q. Mei, “One theme in all views: Modeling consensus topics in multiple contexts,” in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2013, pp. 5–13.

[14] C. Li, Y. Duan, H. Wang, Z. Zhang, A. Sun, and Z. Ma, “Enhancing topic modeling for short texts with auxiliary word embeddings,” ACM Trans. Inf. Syst., vol. 36, no. 2, pp. 1–30, Sep. 2017.

[15] J. Jiang, P. Chen, T. Wang, and X. Wu, “Topic modeling over short texts by incorporating word embeddings,” in Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining, Springer, 2017, pp. 363–374.

[16] W. Liang, R. Feng, X. Liu, Y. Li, and X. Zhang, “GLTM: A global and local word embedding-based topic model for short texts,” IEEE Access, vol. 6, pp. 43612–43621, 2018.

[17] F. Yi, B. Jiang, and I. Wu, “Topic modeling for short texts via word embedding and document correlation,” IEEE Access, vol. 8, pp. 30692–30705, 2020.

[18] T. Shi, K. Kang, J. Choo, and C. K. Reddy, “Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations,” in Proc. World Wide Web Conf., 2018, pp. 1105–1114.

[19] X. Quan, C. Kit, Y. Ge, and S. J. Pan, “Short and sparse text topic modeling via self-aggregation,” in Proc. 24th Int. Joint Conf. Artif. Intell., 2015, pp. 2270–2276.

[20] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, and H. Xiong, “Topic modeling of short texts: A pseudo-document view,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 2105–2114.

[21] Y. W. Teh, “Dirichlet process,” Tech. Rep., 2010.

[22] J. Teh, “Dirichlet process,” Tech. Rep., 2010.

[23] Y. Niu, “Y. Niu et al.: Pitman-Yor Process Self-Aggregated Topic Model for Short Texts of Social Media,” IEEE Access, vol. 6, pp. 43612–43621, 2018.

[24] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” Proc. Nat. Acad. Sci. USA, vol. 101, no. 1, pp. 5228–5233, Apr. 2004.

[25] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, “Learning to classify short and sparse text & web with hidden topics from large-scale data collections,” in Proc. 17th Int. Conf. World Wide Web, 2008, pp. 91–100.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013, arXiv:1301.3781. Available: http://arxiv.org/abs/1301.3781

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013, arXiv:1301.3781. Available: http://arxiv.org/abs/1301.3781

[28] D. Q. Nguyen, R. Binnsley, L. Du, and M. Johnson, “Improving topic models with latent feature word representations,” Trans. Assoc. Comput. Linguistics, vol. 3, no. 1, pp. 299–313, 2015.

[29] R. Das, M. Zaheer, and C. Dyer, “Hierarchical LDA with auxiliary word embeddings,” in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2016, pp. 654–665.

[30] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin, “Automatic evaluation of topic coherence,” in Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics, 2010, pp. 100–108.

[31] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, “Support vector machines,” IEEE Intell. Syst. Appl., vol. 13, no. 4, pp. 18–28, Jul./Aug. 2008.

[32] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in Proc. Int. Joint Conf., Montreal, QC, Canada, Aug. 1995, vol. 14, no. 2, pp. 1137–1145.

[33] J. Tang, M. Zhang, and Q. Mei, “One theme in all views: Modeling consensus topics in multiple contexts,” in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2013, pp. 5–13.

[34] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” Proc. Nat. Acad. Sci. USA, vol. 101, no. 1, pp. 5228–5233, Apr. 2004.

[35] Y. Zuo, J. Zhao, and K. Xu, “Word network topic model: A simple but general solution for short and imbalanced texts,” Knowl. Inf. Syst., vol. 48, no. 2, pp. 379–398, Aug. 2016.

[36] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” Proc. Nat. Acad. Sci. USA, vol. 101, no. 1, pp. 5228–5233, Apr. 2004.

[37] C. Finegan-Dollak, R. Coke, R. Zhang, X. Ye, and D. Radev, “Effects of creativity and cluster tightness on short text clustering performance,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, vol. 1, 2016, pp. 654–665.

YUE NIU received the B.E. degree in software engineering from the Central South University of China. He is currently pursuing the Ph.D. degree in computer science with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include text mining and machine learning.

HONGJIE ZHANG received the B.E. degree in software engineering from Chongqing University. He is currently pursuing the Ph.D. degree in computer science with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include deep reinforcement learning, distributed systems, and cloud computing.

JING LI received the B.S., M.S., and Ph.D. degrees from the University of Science and Technology of China. He is currently a Full Professor with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include machine learning, text mining and reinforcement learning, distributed systems, and cloud computing.

* * *