

TopicOcean: An Ever-Increasing Topic Model With Meta-learning

Yuanfeng Song^{*§}, Yongxin Tong[†], Siqi Bao[‡], Di Jiang[‡], Hua Wu[‡], Raymond Chi-Wing Wong^{*}

^{*}The Hong Kong University of Science and Technology, Hong Kong, China

[†]Beihang University, Beijing, China [‡]Baidu Inc., Beijing, China

^{*}{songyf, raywong}@cse.ust.hk [†]yxtong@buaa.edu.cn [‡]{baosiqi, jiangdi, wu_hua}@baidu.com

Abstract—Topic modeling has been intensively studied and widely applied in both academia and industry in the last decade. In the literature, topic models usually need to be trained from scratch for each individual corpus. Hence, the wisdom of the crowd (i.e., topic models previously trained based upon other corpora) is abandoned. Since a massive amount of in-domain data, considerable computational cost, and human labour are involved in obtaining a high-quality topic model, training from scratch for each new corpus is a huge waste of resources. In this paper, we propose the novel *TopicOcean* framework, which aims to integrate well-trained topic models and transfer the knowledge of accumulated topics to new corpora in order to improve the quality of their topic models. We first propose a method of constructing the ever-increasing TopicOcean, and then propose a meta-learning mechanism that transfers the meta-level knowledge (i.e., topics) in TopicOcean to the scenario of topic modeling on new corpora. Comprehensive experiments validate that the TopicOcean framework can significantly outperform the state-of-the-art (53.77% perplexity improvement on a temporal-shift corpus and 29.24% improvement on a domain-shift corpus). The well-trained high-quality topic models used to construct TopicOcean have been open sourced to promote further research.¹

Keywords—text semantics; topic modeling; meta-learning;

I. INTRODUCTION

Topic modeling is a crucial machine learning technique, which for the past decades has been widely utilized in natural language processing [1], web search [2], recommendation [3] and so on. Recently, advanced methods of training topic models have been introduced, such as AliasLDA [4], LightLDA [5], etc. Despite the achieved progress, to conduct topic modeling for a new corpus, the state-of-the-art methods typically train a new topic model from scratch. Empirically a massive amount of in-domain data and considerable human labour is usually involved in obtaining a high-quality topic model [6]. Considering the great effort made in training a high-quality topic model, it is desirable to develop a framework that can take full advantages of previously well-trained topic models and transfer the knowledge in these models to the scenario of topic modeling on a new corpus, in order to save cost and improve effectiveness.

In this paper, we propose the novel *TopicOcean* framework, with the flowchart of the framework illustrated in

[§]Work conducted while at Baidu Inc.

¹The well-trained topic models can be accessed at Github (https://github.com/baidu/Familia/blob/master/model/download_model.sh)

Figure 1. TopicOcean is inspired by recent work on meta-learning, which focuses on learning that comes from a variety of related tasks and can be used to solve new learning tasks faster and more accurately with limited examples [7]. Conventionally, meta-learning is utilized for supervised learning, and usually involves learning at two levels: higher-level learning conducted across tasks to gain meta-level knowledge and lower-level rapid learning performed for a new task guided by the meta-knowledge learned before [8]. In the unsupervised topic modeling scenario, we utilize meta-learning to emphasize the fact that limited training is involved for model training and topic inference on a new corpus. Two main components are included in the proposed framework: TopicOcean construction and meta-learning based training. In TopicOcean construction, meta-level knowledge (i.e., high-quality topics) is incrementally integrated from models trained on a variety of corpora, and the redundancy of topics is handled on the fly to keep TopicOcean compact. The meta-learning based training comprises two steps: transfer and training. During the transfer step, TopicOcean adopts a Greedy TopicSubset Selection algorithm which transfers its knowledge of TopicOcean as an initialization. In the training step, an efficient inference method is further incorporated to conduct topic model inference with $O(1)$ per word complexity.

The advantages of our proposed framework are essentially twofold. First, TopicOcean significantly reduces the cost of topic modeling on new corpus, which is typically conducted in the literature by training from scratch. Second, as TopicOcean provides readily available topics, practitioners are freed from the laborious training process for each new corpus, and by utilizing its knowledge, this framework avoids generating low-quality topics, which are quite common when parameters are not deliberately calibrated or the volume of the training data is not large enough. To sum up, the contributions of this paper are as follows:

- We have publicly released the four high-quality topic models used to construct TopicOcean. These models are well-trained on industrial-scale datasets and significantly enlarge community's arsenal of topic models.
- We propose the ever-increasing TopicOcean framework to accumulate meta-level knowledge for topic models, and a meta-learning based training algorithm that

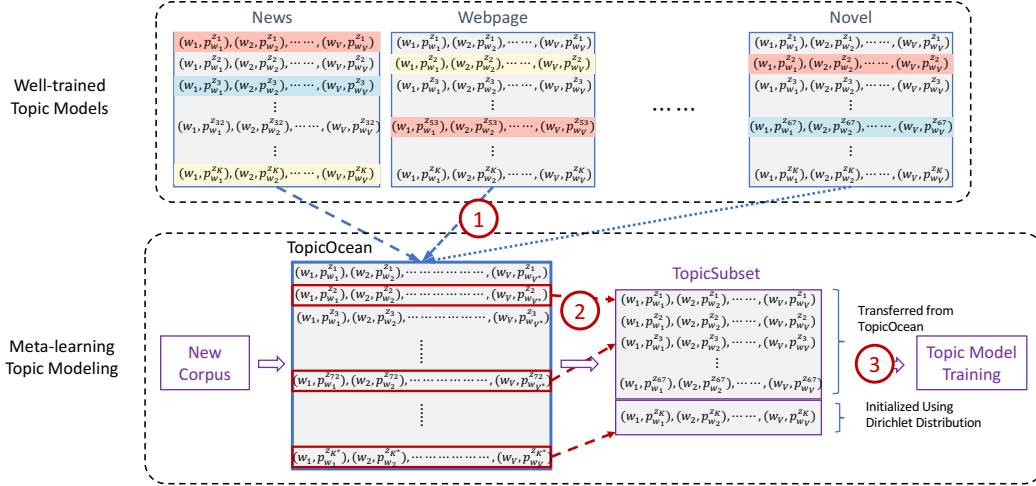


Figure 1. TopicOcean Framework: TopicOcean construction (① incremental TopicOcean construction (redundant topics highlighted with colors)) and Meta-learning model training (② transfer (highlighted with red border), and ③ training for a new corpus)

transfers the knowledge in TopicOcean to new corpora and conducts topic model training and inference with $O(1)$ per word sampling complexity. To the best of our knowledge, this work is the first that studies the problem of topic modeling with a meta-learning technique.

- Comprehensive experiments have been carried out to validate the effectiveness of the TopicOcean framework. Compared with the state-of-the-art, our proposed framework significantly improves topic inference quality up to 53.77% (measured with Perplexity) on a temporal-shift corpus and achieves up to 29.24% improvement on a domain-shift corpus.

The rest of this paper is organized as follows. We review the related work in Section II. Then we introduce the two components of the framework: TopicOcean construction in Section III, and Meta-learning based training in Section IV. The experimental results are presented in Section V, followed by the conclusion in Section VI.

II. RELATED WORK

Meta-learning focuses on learning the experience from a variety of related tasks systematically, and then uses the experience to solve new learning tasks faster and more accurately with limited examples [7]. Based on the types of meta-data these methods leverage, existing approaches can be classified, from the most general to the most task-specific, into three categories according to [9]: learning purely from model evaluations, learning from task properties, and learning from prior models. The learning purely from model evaluations approach aims to learn configurations and search spaces from empirically similar tasks [10]–[12]. The learning from task properties approach learns meta-features or characterizations for each task, and the similarity measurement is defined on the distance between

these features during the knowledge transferring process [13]. The learning from prior models approach is closely related to transfer learning [14], and focuses on transferring trained model parameters or structures between inherently similar tasks. For example, Andrychowicz et al. [15] uses LSTM to predict the gradient for gradient descent algorithms during the model learning period. Reinforcement learning is used in [12] to update the parameters of deep networks and search the network architecture. Model-Agnostic Meta-Learning is proposed in [16] to optimize for a parameter that can quickly adapt to new tasks.

Topic modeling has been recognized as a fundamental tool for searching and understanding large-scale documents. Among all kinds of topic models, Latent Dirichlet Allocation (LDA) [17] is the most commonly used one. Meanwhile, various extensions to LDA have been designed for different applications and in-domain data. For example, Sentence LDA [18] extends LDA by requiring all the words in the same sentence to share the same topic, and this assumption enables superior ability to capture the latent structure of sentences. Although much research has been devoted to designing new topic models by changing their graphical models, the general training paradigm remains the same per se: each topic model needs to be trained from scratch for a new corpus.

III. TOPICOCEAN CONSTRUCTION

In this section, we describe how to construct the ever-increasing TopicOcean. Without loss of generality, we utilize LDA as an example to discuss the technical details.

As shown in Figure 1, a well-trained topic model is stored in a file, with each line having the format of $\{(w_1, p_{w_1}^{z_j}), (w_2, p_{w_2}^{z_j}), \dots\}$, where w_i is the word index, z_j is the topic index and $p_{w_i}^{z_j}$ refers to the normalized weight of

ALGORITHM 1: TopicOcean Construction

```

input :  $\mathcal{M}, \mathcal{M}^N, \delta$ .
output:  $\mathcal{M}^*$ .
1 begin
2   Merge  $\mathcal{M}$  and  $\mathcal{M}^N$  into  $\mathcal{M}^B$ ;
3   Initialize Redundant Topics  $\mathcal{R} = \{\}$ ;
4   for each topic  $z_i$  in  $\mathcal{M}^B$  do
5     for each topic  $z_j$  ( $j > i$ ) in  $\mathcal{M}^B$  do
6       Estimate  $\rho(z_i, z_j)$  with Equation (1);
7       if  $\rho(z_i, z_j) \geq \delta$  then
8         Add  $(z_i, z_j)$  into  $\mathcal{R}$ ;
9    $RS = \text{Union-Find}(\mathcal{R})$ ;
10  for each set  $s$  in  $RS$  do
11    for each topic  $z_{si}$  ( $i > 1$ ) in  $s$  do
12      Add  $z_{si}$  to  $z_{s1}$ ;
13      Remove  $z_{si}$  from  $\mathcal{M}^B$ ;
14   $\mathcal{M}^* = \mathcal{M}^B$ 
15 return  $\mathcal{M}^*$ ;

```

word w_i under topic z_j . Each line in the model represents the word distribution for one topic z_j , denoted as \vec{z}_j .

In order to construct the TopicOcean \mathcal{M} , we need a mechanism that is able to incrementally introduce new topic models \mathcal{M}^N to \mathcal{M} . Since topic models are trained in an unsupervised manner, similar topics exist across different models, as highlighted in Figure 1. In order to keep the compactness of TopicOcean, we need to estimate the similarities and merge similar topics. The detailed algorithm of TopicOcean construction is summarized in Algorithm 1.

For any two topics, we measure the similarity between them and decide whether to merge them or not. Metrics that can be employed includes Weighted Jaccard Similarity [19], Jensen–Shannon Divergence [20] and so on. In this paper, Weighted Jaccard Similarity is utilized because of its relatively good performance in our scenario. Following [21], given two topics z_i and z_j , the similarity is defined as:

$$\rho(z_i, z_j) = \frac{\sum_1^m \min(p_{w_t}^{z_i}, p_{w_t}^{z_j})}{\sum_1^m \max(p_{w_t}^{z_i}, p_{w_t}^{z_j}) + \sum_{m+1}^T p_{w_t}^{z_i} + \sum_{m+1}^T p_{w_t}^{z_j}} = \frac{\sum_1^m \min(p_{w_t}^{z_i}, p_{w_t}^{z_j})}{\sum_1^L p_{w_t}^{z_i} + \sum_1^T p_{w_t}^{z_j} - \sum_1^m \min(p_{w_t}^{z_i}, p_{w_t}^{z_j})}, \quad (1)$$

where P_{z_i} and P_{z_j} are two vectors indicating the top- L words distribution of topic z_i and topic z_j , with detailed definitions $P_{z_i} = (p_{w_1}^{z_i}, p_{w_2}^{z_i}, \dots, p_{w_m}^{z_i}, p_{w_{m+1}}^{z_i}, \dots, p_{w_T}^{z_i})$ and $P_{z_j} = (p_{w_1}^{z_j}, p_{w_2}^{z_j}, \dots, p_{w_m}^{z_j}, p_{w_{m+1}}^{z_j}, \dots, p_{w_T}^{z_j})$. The number m ($0 \leq m \leq L$) represents the number of common words in their top- L words. If the similarity $\rho(z_i, z_j)$ is larger than a pre-defined threshold δ , these two topics are regarded as redundant and need to be merged in the following operations; i.e., the corresponding word of $p_{w_t}^{z_i}$ is the same as that of $p_{w_t}^{z_j}$ for $1 \leq t \leq m$. The numerator sums over the minimum probability for the overlapping words in the two topics, and the denominator sums over the maximum probability for the overlapping words and the probability for the non-overlapping words.

Assuming that three pairs of similar topics are recorded in $\mathcal{R} = \{(z_1, z_2), (z_2, z_3), (z_4, z_5)\}$, z_2 has overlaps with both z_1 and z_3 . If topic merging is carried out independently in each pair, the new topic generated by combining z_1 and z_2 can still be similar to that topic based on z_2 and z_3 . Therefore, instead of topic merging at the pairwise level, we further explore the union sets of redundant topics and carry out topic merging at the set level. We employ the classical Union-Find [22] algorithm to find the disjoint topic sets, and for the above example \mathcal{R} , its disjoint sets are $\{(z_1, z_2, z_3), (z_4, z_5)\}$. For the redundant topics in each set, merging is carried out in the following way: the remaining topics are added to the first topic of this set and are then removed from \mathcal{M}^B . Taking the set (z_1, z_2, z_3) as an example, the topic \vec{z}_1 becomes $\vec{z}_1 + \vec{z}_2 + \vec{z}_3$, with \vec{z}_2 and \vec{z}_3 removed from \mathcal{M}^B . Finally, we can obtain the compact TopicOcean \mathcal{M}^* .

IV. META-LEARNING BASED TRAINING

The meta-learning based training comprises two parts: transfer and training.

A. Transfer

The objective of the *Transfer* is to find a topic subset S from TopicOcean \mathcal{M} as an initialization to boost the following *Training*, such that the TopicSubset S can cover most topics in the new corpus; meanwhile, the redundancy among topics in S is limited. A set function Φ is designed to measure the TopicSubset quality, and TopicSubset selection can be formalized as the following combinatorial optimization problem:

$$S^* \in \underbrace{\arg \max}_{S: S \subseteq \mathcal{M}} \Phi(S) \text{ subject to } |S| \leq m, \quad (2)$$

where S is the TopicSubset, \mathcal{M} is the TopicOcean and m is the number of topics to be selected.

As the objective of TopicSubset selection is two-folds, namely the coverage of topics and the control of redundancy, the quality function Φ is designed as:

$$\Phi(S) = \lambda \mathcal{C}(S) + (1 - \lambda) \mathcal{V}(S), \quad (3)$$

where $\mathcal{C}(S)$ measures the coverage of S for the new corpus, $\mathcal{V}(S)$ rewards diversity inside S , and λ is a trade-off coefficient. Instead of deducting a redundancy term in the quality function, a diversity term (as opposite to redundancy) is added for the convenience of optimization. Detailed definitions and explanations of coverage $\mathcal{C}(S)$ and diversity $\mathcal{V}(S)$ will be given in the following.

The TopicSubset coverage $\mathcal{C}(S)$ is defined as:

$$\mathcal{C}(S) = \sum_{z \in S} c(z), c(z) = \rho(z, \bar{z}), \quad (4)$$

where $c(z)$ is the coverage reward of one single selected topic z . The measurement of $c(z)$ is carried out with the similarity function $\rho(\cdot)$ defined in Equation (1), where \bar{z} is the word distribution of the new corpus and is treated

as a virtual topic. The similarity between one topic z and \bar{z} indicates the coverage reward of single z on the new corpus. By summing over $z \in S$, the total coverage reward of a selected topics can be accumulated in $\mathcal{C}(S)$.

If the quality function Φ only relies on the coverage term (i.e., $\lambda = 1$), it tends to choose those topics whose distributions are similar to the word distribution of the new corpus. In this case, these topics can be selected from some closely-related categories. However, a good topic model is supposed to consist topics from diverse categories; otherwise, its discriminative power would be weakened. As such, the diversity term $\mathcal{V}(S)$ is included in the quality function. Inspired by [23], we define $\mathcal{V}(S)$ as

$$\mathcal{V}(S) = \sum_{i=1}^p \sqrt{\sum_{z \in P_i \cap S} v(z)}, \quad v(z) = \rho(z, \bar{z}), \quad (5)$$

where P_i ($i = 1, \dots, p$) is a partition of TopicOcean \mathcal{M} into separate topic clusters, and $v(z)$ indicates the reward for selecting one topic z from cluster P_i . Through the square root operation, the reward for choosing a topic that is in the same cluster as selected topics will be decreased, thus leading to a more diverse topic selection.

As the combinatorial optimization problem in Equation (2) is known to be NP-complete, we present a Greedy TopicSubset Selection (GTS) algorithm to solve it, as illustrated in Algorithm 2. The algorithm first separates the TopicOcean \mathcal{M} into several partitions using conventional clustering methods (Line 3). Then it searches over all the topics in TopicOcean \mathcal{M} and each time selects the one with the maximal incremental Φ value (Lines 5–6) until it finds m topics. According to the aforementioned definitions, it is easy to conclude that $\mathcal{C}(S)$ and $\mathcal{V}(S)$ are both monotone nondecreasing submodular functions. Therefore, Φ is a monotone submodular function as well. Let S_{GTS} be the set returned by the GTS algorithm and S^* be the optimal set. If Φ is a monotone nondecreasing submodular function [24], then we have the worst-case bound as follows:

$$\Phi(S_{GTS}) \geq (1 - \frac{1}{e})\Phi(S^*). \quad (6)$$

B. Training

The meta-learning based training is formally presented in Algorithm 3. The algorithm first selects a TopicSubset that contains m topics from TopicOcean using the GTS described in Algorithm 2 (Line 2), and the other $K - m$ topics are initialized according to Dirichlet distribution just like the conventional LDA model (Line 3). Then the selected topics and the newly initialized topics are composed together as the topic model under training (Line 4).

According to [25], Gibbs sampling calculates the following conditional probability for all topics and performs normalization:

$$p(z_{di} = k | \text{rest}) \propto (n_{kd}^{-di} + \alpha_k) \frac{(n_{kw} + \beta_w)}{n_k + \beta}, \quad (7)$$

ALGORITHM 2: Greedy TopicSubset Selection

input : Topic Partition Number p , Topic Number of TopicSubset m , TopicOcean \mathcal{M} , New Corpus \mathcal{N} , Parameter λ .
output: TopicSubset S .

```

1 begin
2    $S = \{\}$ ;
3   Separate  $\mathcal{M}$  into  $p$  topic partitions using clustering methods;
4   while  $|S| < m$  do
5      $z = \arg \max_{\bar{z} \in \mathcal{M} - S} (\Phi(\{\bar{z}\} \cup S) - \Phi(S))$ ;
6      $S = S + \{z\}$ ;
7     if  $\mathcal{M} - S = \emptyset$  then
8       break;
9 return TopicSubset  $S$ .
```

where z_{di} is the topic assignment of word i in document d ; n_{kd}^{-di} is the number of times that d is assigned to topic k through multinomial distribution, except current word i ; α_k , β_w and β are hyperparameters for Dirichlet distribution, and $\frac{(n_{kw} + \beta_w)}{n_k + \beta}$ represents the probability of word w appearing in topic k based on the statistics of the selected TopicSubset.

Since the above time complexity to sample a topic for a word is $O(K)$ (K is the topic amount of the TopicSubset), we use an efficient alternative based upon Metropolis Hastings (MH) [5] to speed up the topic inference, which achieves $O(1)$ per word time complexity. We first conduct one approximation to Equation (7) in the following way:

$$q(z_{di} = k | \text{rest}) \propto \underbrace{(n_{kd} + \alpha_k)}_{\text{document-based proposal}} \underbrace{\frac{(n_{kw} + \beta_w)}{n_k + \beta}}_{\text{word-based proposal}}. \quad (8)$$

In order to achieve high sampling performance, we can accelerate the sampling process by well-designed proposals, namely, document-based proposal and word-based proposal for MH, as follows:

- Document-based Proposal:

$$p_d(k) \propto \frac{n_{dk} + \alpha_k}{\sum_{k'=1}^K (n_{dk'} + \alpha_{k'})}. \quad (9)$$

According to the MH algorithm, the acceptance ratio from state i to j is:

$$\min \{1, \frac{p(j)p_d(i)}{p(i)p_d(j)}\}. \quad (10)$$

- Word-based Proposal:

$$p_w(k) \propto \frac{n_{kw} + \beta_w}{\sum_{u=1}^U (n_{ku} + \beta_u)}, \quad (11)$$

where U is the word number of topic k . The acceptance ratio from state i to j is:

$$\min \{1, \frac{p(j)p_w(i)}{p(i)p_w(j)}\}. \quad (12)$$

In TopicOcean, we resort to the alias method [26] to reduce the sampling complexity. The alias method mainly relies on a data structure named an alias table. Each word is represented as the probability distribution over distinct topics, and the corresponding alias table for each word is

ALGORITHM 3: Meta-learning based Training

input : Topic Partition Number p , Total Topic Number K ,
Transferred Topic Number m , TopicOcean \mathcal{M} , New
Corpus \mathcal{N} , λ .

output: Topic Distribution for each d in \mathcal{N} .

```

1 begin
2    $T_s = GTS(p, m, \mathcal{M}, \mathcal{N}, \lambda)$ ;
3    $T_o =$  Initialize the rest  $K - m$  topic distribution  $\sim Dir(\beta)$ ;
4    $T = T_s + T_o$ ;
5   Build alias tables using  $T$ ;
6   Initialize a topic  $z$  for each word  $w$  in  $\mathcal{N}$  based on alias
   table of  $w$ ;
7   for each document  $d \in \mathcal{N}$  do
8     propose a topic  $z_d$  based on document-topic proposal
     according to Equation (9);
9     update the topic to  $z'_d$  according to acceptance ratio by
     Equation (10);
10    for each word  $w$  in  $d$  do
11      propose a topic  $z_w$  based on alias table of  $w$ 
      according to Equation (11);
12      update the topic to  $z'_w$  according to acceptance ratio
      by Equation (12);
13      if  $z_w \in T_o$  then
14        update  $z'_w$  and  $z_w$ 's topic-word distribution;
15    Calculate the topic distribution for each  $d$  in  $\mathcal{N}$ ;
16 return The topic distribution for each  $d$  in  $\mathcal{N}$ ;

```

constructed for the acceleration of subsequent topic inference (Line 5). When sampling a new topic for each word, the algorithm iteratively utilizes a document-based proposal and word-based proposal to update the topic candidates (Line 11). For each topic candidate, the algorithm chooses whether to accept it according to the acceptance ratio (Line 12). Since the topics selected from the TopicOcean are well-trained, no topic-word distribution updating is required for them, and the algorithm only updates the new topics (Line 13 and 14). This sampling procedure is repeated for several iterations until convergence.

V. EXPERIMENTS

In this section, we evaluate the performance of the TopicOcean framework in terms of two quantitative metrics.

A. Experimental Setup

We construct TopicOcean by integrating four industrial-scale well-trained topic models, which are trained on distinct corpora. Specifically, the four topic models, denoted as M_1, M_2, M_3 and M_4 , are trained with large-scale datasets of Chinese news, webpages, novels and weibo posts², respectively. The statistics of the topic models are presented in Table I. To evaluate the performance of the proposed TopicOcean, extensive experiments have been carried out on two new corpora – Weibo2014 and Ads. Weibo2014 includes 100k weibo posts crawled in 2014 and Ads consists of 200k advertisements crawled from one commercial search engine. All programs are written in Python, and all experiments are

²<https://www.weibo.com>, data format similar to Tweet.

performed on a server with 128GB memory, 16 Intel Core Processor (Haswell), and CentOS.

Table I
DETAILS OF FOUR WELL-TRAINED TOPIC MODELS.

Topic Models	# of Topics	Data Type	Data Scale	Vocabulary Size
M_1	2000	News	Tens of millions	294,657
M_2	4267	Webpages	Tens of millions	283,827
M_3	500	Novels	Hundreds of thousands	243,617
M_4	2000	Weibo	Hundreds of millions	175,347

B. TopicOcean Statistics

The mission of TopicOcean construction is to obtain a comprehensive and compact topic model, which is not sensitive to the merge order. In the experiments, TopicOcean \mathcal{M} is set to M_1 as initialization. The other topic models M_2, M_3 and M_4 are incrementally integrated into TopicOcean according to Algorithm 1. In our experiments, top-L is set to 30 for topic similarity estimation, and the similarity threshold δ is set to 0.4 empirically. The process of constructing TopicOcean is shown in Figure 2. M_2, M_3 and M_4 are added into TopicOcean in an incremental way for the sake of computational efficiency, as represented by the three sets of bars. Finally, we obtain a compact TopicOcean containing 7549 topics in total.

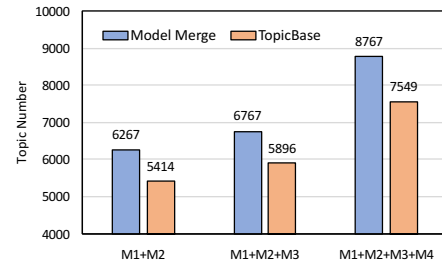


Figure 2. TopicOcean Statistics.

C. Comparison of Perplexity

The experiments are carried out with two corpora – Weibo2014 and Ads, which have distinct temporal phases and domains, respectively, with the data utilized to construct TopicOcean. With these temporal-shift (Weibo2014) and domain-shift (Ads) data, we dissect the transfer ability of meta-learning based training in detail.

The perplexity on each corpus is displayed in Figure 3, with a range of topic number settings $K = 200$ and number of transferred topics $M \in \{0, 5\%K, \dots, 90\%K\}$. Note that the lower the perplexity that a method obtains, the better its performance is. As shown in Figure 3, *MLTM* methods can obtain the better performance across distinct corpora and with different topic number settings in most cases. Specially, *MLTM* can transfer 90% of the topics from TopicOcean, and achieve 53.77% improvement in perplexity compared with

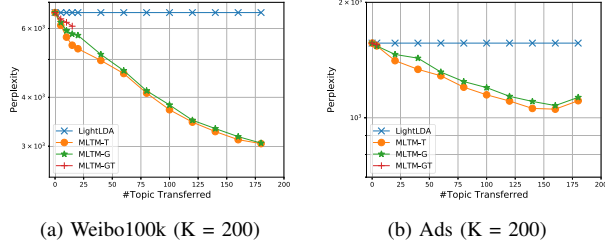


Figure 3. Effectiveness Comparison: (a) Temporal-shift Quality Evaluation and (b) Domain-shift Quality Evaluation.

the baseline. In the more challenging scenario — domain shift evaluation, as shown in Figure 3b — *MLTM* can also transfer up to 80% of the topics from TopicOcean, and achieve 29.24% improvement in perplexity.

D. Comparison of Coherence

The topic coherence measurement reflects the degree of semantic similarity between frequent words in the topic. We use the averaged topic coherence [27] to evaluate the quality of the topic models produced by *LightLDA* and *MLTM*, as presented in Table II. The Wikipedia dataset is used as the reference corpus for coherence calculation. Top- L ($L \in \{5, 10\}$) defines the number of words we consider to calculate coherence for each topic. It can be observed that *MLTM* provides higher-quality models than *LightLDA*. Specifically, in the Ads dataset, *MLTM-G* achieves 65.98% improvement for top-5 and 45.56% for top-10 compared with the baseline in terms of coherence.

Table II
AVERAGED TOPIC COHERENCE BY LIGHTLDA AND MLTM.

Method	Ads(K=200 N=160)		Weibo100k(K=200 N=180)	
	Top-5	Top-10	Top-5	Top-10
LightLDA	0.2737	0.2307	0.2576	0.2323
MLTM-G	0.4543	0.3358	0.4298	0.3228
MLTM-T	0.4523	0.3257	0.4065	0.3049

VI. CONCLUSION

In this paper, the ever-increasing TopicOcean with a meta-learning based training algorithm is proposed to solve the problems plaguing conventional topic modeling. Within this framework, the ever-increasing TopicOcean is constructed to incrementally integrate topics from previous well-trained topic models. Based upon TopicOcean, a novel method is proposed to adapt TopicOcean to new corpora and conduct topic inference. Experimental results demonstrate that TopicOcean can outperform its counterpart with better quality and higher efficiency. In the future, we plan to apply TopicOcean to more downstream NLP tasks.

ACKNOWLEDGEMENT

This research is partially supported by ITS/227/17FP. We are grateful to the anonymous reviewers for their constructive comments on this paper.

REFERENCES

- [1] R. Das, M. Zaheer, and C. Dyer, “Gaussian lda for topic models with word embeddings,” in *ACL*, 2015.
- [2] W. Song, Y. Zhang, T. Liu, and S. Li, “Bridging topic modeling and personalized search,” in *Coling*, 2010.
- [3] C. Wang and D. M. Blei, “Collaborative topic modeling for recommending scientific articles,” in *KDD*. ACM, 2011.
- [4] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, “Reducing the sampling complexity of topic models,” in *KDD*, 2014.
- [5] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T.-Y. Liu, and W.-Y. Ma, “Lightlda: Big topic models on modest computer clusters,” in *WWW*, 2015.
- [6] D. Jiang, Y. Song, R. Lian, S. Bao, J. Peng, H. He, and H. Wu, “Familia: A Configurable Topic Modeling Framework for Industrial Text Engineering,” *arXiv preprint arXiv:1808.03733*, 2018.
- [7] Z. Li, F. Zhou, F. Chen, and H. Li, “Meta-sgd: Learning to learn quickly for few shot learning,” *arXiv preprint arXiv:1707.09835*, 2017.
- [8] F. Zhou, B. Wu, and Z. Li, “Deep meta-learning: Learning to learn in the concept space,” *arXiv preprint arXiv:1802.03596*, 2018.
- [9] J. Vanschoren, “Meta-learning: A survey,” *arXiv preprint arXiv:1810.03548*, 2018.
- [10] S. M. Abdulrahman, P. Brazdil, J. N. van Rijn, and J. Vanschoren, “Speeding up algorithm selection using average ranking and active testing by introducing runtime,” *Machine learning*, vol. 107, no. 1, pp. 79–108, 2018.
- [11] H. Weerts, M. Meuller, and J. Vanschoren, “Importance of tuning hyperparameters of machine learning algorithms,” Tech. rep., TU Eindhoven, Tech. Rep., 2018.
- [12] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” in *ICLR*, 2017.
- [13] A. Rivolli, L. P. Garcia, C. Soares, J. Vanschoren, and A. C. de Carvalho, “Towards reproducible empirical research in meta-learning,” *arXiv preprint arXiv:1808.10406*, 2018.
- [14] S. J. Pan and Q. Yang, “A survey on transfer learning,” *TKDE*, 2010.
- [15] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, “Learning to learn by gradient descent by gradient descent,” in *NIPS*, 2016.
- [16] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” *arXiv preprint arXiv:1703.03400*, 2017.
- [17] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, 2003.
- [18] G. Balikas, M.-R. Amini, and M. Clausel, “On a topic model for sentences,” in *SIGIR*. ACM, 2016.
- [19] F. Chierichetti, R. Kumar, S. Pandey, and S. Vassilvitskii, “Finding the jaccard median,” in *SODA*. SIAM, 2010.
- [20] B. Fuglede and F. Topsøe, “Jensen-shannon divergence and hilbert space embedding,” in *ISIT*. IEEE, 2004.
- [21] D. Jiang, Y. Song, Y. Tong, X. Wu, W. Zhao, Q. Xu, and Q. Yang, “Federated topic modeling,” in *CIKM*, 2019.
- [22] Z. Galil and G. F. Italiano, “Data structures and algorithms for disjoint set union problems,” *ACM Computing Surveys (CSUR)*, vol. 23, no. 3, pp. 319–344, 1991.
- [23] H. Lin and J. Bilmes, “A class of submodular functions for document summarization,” in *ACL*, 2011.
- [24] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, “An analysis of approximations for maximizing submodular set functions—i,” *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.
- [25] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” *Proceedings of the National Academy of Science*, vol. 101, pp. 5228–5235, Apr. 2004.
- [26] G. Marsaglia, W. W. Tsang, J. Wang *et al.*, “Fast generation of discrete random variables,” *Journal of Statistical Software*, 2004.
- [27] M. Röder, A. Both, and A. Hinneburg, “Exploring the space of topic coherence measures,” in *WSDM*, 2015.