Leveraging Meta Information in Short Text Aggregation

He Zhao† Lan Du†∗ Guanfeng Liu‡ Wray Buntine†
†Faculty of Information Technology
Monash University, Australia
ethanhezhao@gmail.com, {lan.du, wray.buntine}@monash.edu
‡Department of Computing
Macquarie University, Australia
guanfeng.liu@mq.edu.au

Abstract

Analysing topics in short texts (e.g., tweets and new headings) is a challenging task because short texts often contain insufficient word co-occurrence information, which is important to learn good topics in conventional topic topics. To deal with the insufficiency, we propose a generative model that aggregates short texts into clusters by leveraging the associated meta information. Our model can generate more interpretable topics as well as document clusters. We develop an effective Gibbs sampling algorithm favoured by the fully local conjugacy in the model. Extensive experiments demonstrate that our model achieves better performance in terms of document clustering and topic coherence.

1 Introduction

Texts generated on the internet (e.g., tweets, news headlines and product reviews) are usually short, which means that each individual document contains insufficient word co-occurrence information. Many existing topic models like Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and its variants infer topics purely based on the word occurrence information, which often results in degraded performance and makes those models incapable of learning from short texts.

Recently, many research efforts have been devoted to analysing short texts. A common strategy is to aggregate short texts into clusters and then apply topic models to those clusters. The clusters are expected to aggregate the word co-occurrence information of the assigned documents. One widely-used option is known as self-aggregation, where we can aggregate short texts according to the contextual information. For example, the contextual information of a document can be encoded by its topics so that the topic assignments can be used for aggregation. This line of research includes models such as SATM (Quan et al., 2015), LTM (Li et al., 2018a), and PTM (Zuo et al., 2016a). On the other hand, many short texts, likes tweets, often come with meta information (meta-info for short, also known as meta-data or side information), such as authors, categories, hashtags, timestamps, etc. Therefore, another popular option is to aggregate short texts according to their meta-info. For example, we can assume that tweets published by the same users (Hong and Davison, 2010; Zhao et al., 2011) or with the same hashtags (Mehrotra et al., 2013) are likely to discuss similar topics. Those tweets can be aggregated into the same clusters.

Although the above two aggregation schemes have yielded prominent results on short text analysis, there is still space for improvement. For example, in tweet analysis: if we ignore the associated meta-info as in the self-aggregation scheme, we may lose important information; on the other hand, it may not be a perfect idea to simply aggregate the tweets according to one kind of its meta-info such as hashtags, because the amounts of the tweets in different hashtags may differ largely and the diversity of the tweets labelled by one hashtag can be dramatic. In this paper, we are interested in developing a principle way of incorporating the meta-info directly into the generative process of a self-aggregation model, so that we can take advantage of both aggregation schemes in one integrated model. Here we present the Meta-Info Guided Aggregation (MIGA) model, a new self-aggregation model whose aggregation process is guided by the meta-info associated with each individual short text. Specifically, MIGA aggregates short texts according to two factors: whether those texts have similar content and whether they share similar meta-info. The proposed model assumes that the more short texts share the meta-info and discuss similar topics, the more likely they are as-

Corresponding author
signed to the same cluster. Moreover, MIGA automatically balances the two factors in a principled way. The flexibility in the framework of MIGA also allows us to leverage hierarchical meta-info and/or the pre-trained word embeddings to further improve the model performance.

2 Related Work

In addition to the aforementioned aggregation or pooling based models, another popular research direction for short text topic modelling is using word correlations or embedding to enhance topic models. For example, Biterm Topic Model (BTM) (Yan et al., 2013) and Relational BTM (Li et al., 2018b) enrich each document with word pairs (i.e., biterms). Instead of using all the word pairs, Yang et al. (2015a) considers pre-trained phrases. While Word Network Topic Model (WNTM) (Zuo et al., 2016b) leverages a network of word co-occurrences. Yang et al. (2018) uses document-level co-occurrence patterns. With word embeddings, Latent Feature LDA (LFLDA) (Nguyen et al., 2015) mixes a Dirichlet-Multinomial model with a softmax function of word embeddings; Word-Topic Mixture (WTM) model (Fu et al., 2016) combines the idea of LFLDA and Topical Word Embedding (TWE) model (Liu et al., 2015); Gaussian LDA (GLDA) (Das et al., 2015) directly generates topic-word distributions, respectively. Without utilising word semantic relations computed from pre-trained word embeddings, MetaLDA (Zhao et al., 2017c, 2018a), WEFTM (Zhao et al., 2017b), and WEDTM (Zhao et al., 2018c) leverage binary and real-valued word embeddings in the topic-word distributions, respectively. Without using external word embeddings, DirBN (Zhao et al., 2018b) can be viewed as a self-aggregation model which aggregates the word co-occurrence information with a multi-layer structure.

The proposed model, MIGA, falls into the category of aggregation based models. Compared with others in this line, the major novelty of MIGA is that it considers both meta-info and content of short texts in the aggregation process, while existing models only take one factor into account. For short text models with word embeddings, they may face problems when the contextual information of the external word embeddings is not consistent with the contextual information in the target corpus. For example, the word embeddings trained on large general corpora may not be suitable for a specialised target corpus. Compared with those models, MIGA does not rely on external information of words. Moreover, if applicable, MIGA can also be flexibly extended with hierarchical meta-info and word embeddings.

3 The Proposed Model

Given a set of $D$ short documents, the existing self-aggregation methods such as PTM (Zuo et al., 2016a) assume that each document $d \in \{1, \ldots, D\}$ belongs to one of $M$ latent clusters. Each cluster accumulates the word counts from the assigned documents and contains more sufficient word co-occurrence information than an individual document. To generate the $i$th ($i \in \{1, \ldots, N_d\}$) word $w_{d,i}$ in document $d$ with $N_d$ words, we first sample $d$’s cluster assignment $c_d = m \in \{1, \ldots, M\}$ according to its $doc$-cluster distribution, i.e., $\psi_d \in \mathbb{R}_+^M$; and then we sample a topic $z_{d,i} \in \{1, \ldots, K\}$ for word $w_{d,i}$ from the $cluster$-$topic$ distribution $\theta_{c_d} \in \mathbb{R}_+^K$. Given $z_{d,i}$, $w_{d,i}$ is sampled from the $topic$-$word$ distribution $\phi_{z_{d,i}} \in \mathbb{R}_+^V$, where $V$ is the size of the vocabulary in the target corpus.

Different from the existing self-aggregation methods, which impose an uninformative prior on $\psi_d$, our model draws it from a document-specific prior $\pi_d \in \mathbb{R}_+^M$, constructed from $d$’s meta-info. Assume that there are $L$ unique labels2 in a corpus and the labels of document $d$ are encoded in a binary vector $f_d \in \{0, 1\}^L$, where $f_{d,l} = 1$ indicates $d$ has label $l$. This encoding method allows a

---

2The inference algorithm of the proposed model is elaborated in the appendix.

3Hereafter, we use “labels” and “meta-info” interchangeably, even though our model is able to incorporate any meta-info in discrete formats.
document to have multiple labels. Figure 1 shows the full generative process of MIGA, which is also described as follows:

1. For each latent cluster $m$:
   (a) For each label $l$: $\lambda_{l,m} \sim \text{Ga}(\mu_0, \mu_0)$
   (b) Draw: $\theta_m \sim \text{Dir}_K(\alpha)$
2. For each topic $k$, draw $\phi_k \sim \text{Dir}_V(\beta_0)$
3. For each document $d$:
   (a) For each cluster $m$, compute: $\pi_{d,m} = \prod_{l=1}^{L}(\lambda_{l,m})_{d,l}$
   (b) Draw: $\psi_d \sim \text{Dir}_L(\pi_d)$
   (c) Draw the cluster assignment: $c_d \sim \text{Cat}_M(\psi_d)$
4. For each word $i$ in document $d$:
   (a) Draw topic: $z_{d,i} \sim \text{Cat}_K(\theta_{c_d})$
   (b) Draw word: $w_{d,i} \sim \text{Cat}_V(\phi_{z_{d,i}})$

where $\text{Ga}(\cdot, \cdot)$ is the gamma distribution with the shape and rate parameters; $\text{Dir}_K(\cdot)$ is the $K$ dimensional Dirichlet distribution; $\text{Cat}_K(\cdot)$ is the $K$ dimensional categorical distribution.

The main idea of MIGA is the meta-info guided aggregation, where instead of putting an uninformative prior on $\psi_d$, MIGA constructs an informative document-specific Dirichlet prior with parameter $\pi_d$ computed from the document’s labels. Specifically, in Step 3a above, $\lambda_{l,m}$ captures the correlations between cluster $l$ and cluster $m$. If document $d$ has label $l$, i.e., $f_{d,l} = 1$, $\lambda_{l,m}$ contributes to $\pi_{d,m}$, which is the prior of $\psi_{d,m}$. This shows how the meta-info influences the probability of assigning a document to a cluster. Moreover, in our model, meta-info only contributes to the prior and the actual value of $\psi_{d,m}$ is eventually determined by both the prior and the evidence (i.e., the content of $d$), according to Bayes’ theorem. The incorporation of meta-info in the Dirichlet prior of our model is related to the ones in Zhao et al. (2017a, 2018d), but theirs work in different domains.

**Leveraging hierarchical meta-info:** MIGA can be extended to accommodate hierarchical meta-info (e.g., an academic paper labelled with tags “computer science→machine learning→deep learning”). Let us consider a two-layer hierarchy, where the $L$ document labels (i.e., the first-layer labels) are further categorized into a set of $L'$ super classes (i.e., the second-layer labels). Note that one document label is allowed to belong to multiple super classes and $f_{l',m} \in \{0, 1\}$ is used to denote whether or not a first-layer label $l$ belongs to a second-layer label $l'$. The general idea here is that instead of drawing $\lambda_{l,m}$ from an uninformative gamma prior as in our original model, we draw it from a prior distribution informed by the second-layer labels, as follows:

$$\lambda_{l',m} \sim \text{Ga}(\mu_0, \mu_0),$$

$$\lambda_{l,m} \sim \text{Ga}\left(\prod_{l'=1}^{L'} (\lambda_{l',m})_{f_{l',m}}^{f_{l',m}}, \mu_0\right),$$

where $\lambda_{l,m}$ captures the correlation between labels at the two layers. Thus, the information of the second-layer labels will be propagated down to the assignment process of the documents.

**Leveraging word embeddings:** MIGA can be extended to incorporate word embeddings to guide the generation of latent topics. Following the approach introduced in Zhao et al. (2017c), we draw $\phi_k \sim \text{Dir}_V(\beta_k)$, where $\beta_k \in \mathbb{R}^V_+$ is computed with a log-linear model of word embeddings, similar to Step 3a in the generative process.

### 4 Experiments

We evaluate the performance of MIGA on document clustering and topic coherence, with several advances in short text topic modelling\(^3\). We also provide a set of qualitative analysis to demonstrate the interpretability of our model.

The details of the datasets used in the experiments are shown in Table 1. The hyper-parameter settings of the proposed model are as follows: For MIGA, we set $\beta_0 = 0.01, \mu_0 = 1.0$, and imposed gamma prior on each entry of $\alpha$; for the

| Dataset       | $D$  | $V$  | avg $N_d$ | $L$  |
|---------------|------|------|-----------|------|
| Tweets (Mehrotra et al., 2013) | 87,608 | 24,884 | 11 | 6 |
| Patents\(^4\) | 13,588 | 3,745 | 9 | $L' = 4$ |
| Web Snippets (Li et al., 2016) | 12,237 | 10,052 | 15 | 8 |
| Stackoverflow (Xu et al. 2017) | 18,287 | 2,458 | | $L' = 6$ |
| 20Newsgroups\(^5\) | 10,020 | 2000 | 28 | $L' = 20$ |

\(^3\)We also conducted text classification experiments with the compared models but MIGA did not show significant improvements over MetaLDA.

\(^4\)Collected from https://www.lens.org.

\(^5\)http://qwone.com/˜jason/20Newsgroups/. A subset with the most frequent 2000 vocabulary words and the documents with less than 50 words was used.
other models, we used their best settings reported in their papers unless otherwise specified. The number of MCMC iterations for training was set to 2000 for all the models.

**Document Clustering:** To measure the clustering performance we used purity and Normalized Mutual Information (NMI), which are commonly used metrics for clustering (Manning et al., 2008). We compared MIGA with PTM (Zuo et al., 2016a), MetaLDA (Zhao et al., 2017c), GPUDMM (Li et al., 2016, 2017), as well as KMeans with different document features. The experiment was run as follows: 80% documents in each dataset were used in training, purity was computed on the remaining documents; For MetaLDA and MIGA, the labels of the test documents were excluded in testing for fair comparison; For PTM and MIGA, we set \( K = 50 \) and varied \( M \), and the cluster assignments were used in computing the two scores; For LDA, MetaLDA, and GPUDMM, we used the topic with the largest weight as the cluster assignment, i.e., the number of topics equals to the number of clusters, following Nguyen et al. (2015); KMeans with two kinds of document features (\( V \) dimensional TFIDF vectors and \( K = 50 \) dimensional document-topic vectors extracted from LDA) served as the baselines. Table 2 shows the purity and NMI scores.\(^7\)

| Dataset    | Tweets | Patents | Web Snippets | Stackoverflow | 20Newsgroups |
|------------|--------|---------|--------------|---------------|--------------|
| #Clusters  | 100    | 200     | 500          | 100 200 500   | 100 200 500  |
| KMeans + TFIDF | -      | -       |   -          | 0.36 04 0.51  | 0.49 0.60 0.73 |
| KMeans + LDA  | 0.69   | 0.70    | 0.72         | 0.50 0.52 0.57| 0.69 0.71 0.76 |
| MetaLDA     | 0.70   | 0.70    | 0.64         | 0.57 0.55 0.49| 0.77 0.82 0.82 |
| GPUDMM      | 0.59   | 0.62    | 0.63         | 0.36 0.37 0.41| 0.69 0.71 0.72 |
| PTM         | 0.84   | 0.83    | 0.84         | 0.53 0.53 0.55| 0.61 0.68 0.77 |
| MIGA        | 0.92   | 0.93    | 0.92         | 0.57 0.59 0.61| 0.71 0.77 0.80 |

### (a) Purity

| Dataset    | Tweets | Patents | Web Snippets | Stackoverflow | 20Newsgroups |
|------------|--------|---------|--------------|---------------|--------------|
| #Clusters  | 100    | 200     | 500          | 100 200 500   | 100 200 500  |
| KMeans + TFIDF | -      | -       |   -          | 0.16 0.20 0.23| 0.29 0.34 0.39 |
| KMeans + LDA  | 0.26   | 0.25    | 0.25         | 0.21 0.22 0.25| 0.35 0.35 0.36 |
| MetaLDA     | 0.29   | 0.26    | 0.23         | 0.25 0.24 0.19| 0.41 0.42 0.41 |
| GPUDMM      | 0.21   | 0.23    | 0.24         | 0.10 0.12 0.15| 0.36 0.38 0.38 |
| PTM         | 0.39   | 0.36    | 0.35         | 0.23 0.24 0.25| 0.29 0.33 0.38 |
| MIGA        | 0.47   | 0.45    | 0.43         | 0.26 0.27 0.29| 0.35 0.39 0.40 |

### (b) NMI

MIGA outperforms the other models on Tweets, Patents, and Stackoverflow, which are relatively shorter than the other datasets. This demonstrates our model’s effectiveness on clustering short texts.

**Topic Coherence:** Topic coherence measures the semantic coherence in the most significant (top) words in a topic, which is another commonly-used metric for topic models. Here we used the Normalized Pointwise Mutual Information (NPMI) (Aletras and Stevenson, 2013; Lau et al., 2014) to calculate a topic coherence score of the top 10 words of each topic.\(^8\) Following Yang et al. (2015b), to eliminate rare topics, we report the scores over the top 50% topics with the largest number of words (i.e., for a topic \( k \), we can count the number of words that are assigned to it: \( \sum_{d=1}^{D} \sum_{i=1}^{N_d} 1_{z_{d,i}=k} \)). It is known that word embeddings are able to significantly improve topic coherence. Therefore, in this experiment, following Zhao et al. (2017c), we used the word embeddings binarised from the pre-trained 50-dimensional GloVe word embeddings (Pennington et al., 2014) in MIGA and MetaLDA, denoted as MIGA-eb and MetaLDA-eb, respectively. For all the models, we set \( K = 50 \). For PTM, MIGA, and MIGA-eb, we report the best scores with \( M \) varying from 100 to 3000. Table 4 shows the NPMI scores, where in general, among the models without word embeddings, MIGA outperforms the others on most datasets. Moreover, word embeddings

\(^7\)Original MetaLDA is able to use both document meta-info and word embeddings. Here we used its variant only with document meta-info.

\(^8\)We used the Palmetto package (http://palmetto.aksw.org) with a large Wikipedia dump to compute NPMI.
Qualitative Analysis: Table 3 shows the relations between the hierarchical labels in 20Newsgroups. We started with a second-layer label “comp” and found its most related topics by first selecting the most related clusters (by ranking $\lambda_{l,m}^{m}$) then selecting the most related topics (by ranking $\theta_{m,k}$). Next, we looked at the first-layer labels (marked in italic) associated with $l$, i.e. $f_{l,l'} = 1$ and then found the most related topics in a similar way. Right: Clusters, documents, and topics for the label of “Computers” on Web Snippets, discovered by MIGA with $K = 100, M = 500$. We first selected the most related clusters to “Computers” by ranking $\lambda_{l,m}$, and then selected the most related documents and the most related topic in each cluster by ranking $\pi_{d,m}$ and $\theta_{m,k}$, respectively.

Table 3: Left: Topics related to the hierarchical labels in 20Newsgroups. We started with a second-layer label “comp” and found its most related topics by first selecting the most related clusters (by ranking $\lambda_{l,m}^{m}$) then selecting the most related topics (by ranking $\theta_{m,k}$). Next, we looked at the first-layer labels (marked in italic) associated with $l$, i.e. $f_{l,l'} = 1$ and then found the most related topics in a similar way. Right: Clusters, documents, and topics for the label of “Computers” on Web Snippets, discovered by MIGA with $K = 100, M = 500$. We first selected the most related clusters to “Computers” by ranking $\lambda_{l,m}$, and then selected the most related documents and the most related topic in each cluster by ranking $\pi_{d,m}$ and $\theta_{m,k}$, respectively.

Table 4: NPMI for topic coherence. The best scores are in boldface.

Further help improve topic coherence of MIGA-eb. It is noteworthy that MIGA and MIGA-eb do not improve NPMI over PTM on Tweets. There are two possible factors: the labels of the tweets are not informative enough for MIGA to learn better clusters and the vocabulary of this dataset consists many slangs and abbreviations, which are not included in the corpus used for calculating NPMI.

Qualitative Analysis: The left sub-table of Table 3 shows the relations between the hierarchical labels and topics discovered by MIGA in 20Newsgroups. One can see that the associated topics of the second-layer document label, “comp”, are more general ones, describing several general aspects of computers, while the topics associated with the first-layer labels are relatively more specific. For example, the associated topics of “comp.sys.ibm.pc.hardware” are specific ones describing different aspects of computer hardware. The right sub-table shows the relations between clusters, documents, and topics discover by MIGA in Web Snippets. It can be observed that the documents in Web Snippets labelled with “Computers” are quite diverse, which can be further clustered into the ones related to “hardware”, “digital products”, “programming language”, and so on. Therefore, simply aggregating those documents into one cluster as in previous meta-info aggregation models may not be appropriate. MIGA can discover fine-grained latent clusters, each of which focuses on different aspects of “Computers” and can intuitively be interpreted by its top topic.

5 Conclusion

We have presented a new aggregation framework, MIGA, for short text topic analysis. MIGA is able to aggregate short text documents into latent clusters by leveraging meta-info. MIGA takes advantages of previous models which perform aggregation according to either content or meta-info in short texts. The proposed framework can be easily extended with hierarchical meta-info and word embeddings. The experimental results have shown that MIGA achieves improved performance on document clustering, topic coherence, as well as appealing interpretability. For future study, we would like to investigate how to automatically learn the number of latent clusters with non-parametric Bayesian methods.
References

Nikolaos Aletras and Mark Stevenson. 2013. Evaluating topic coherence using distributional semantics. In International Conference on Computational Semantics, pages 13–22.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet allocation. *JMLR*, pages 993–1022.

W. Buntine and M. Hutter. 2012. A Bayesian view of the Poisson-Dirichlet process. arXiv preprint arXiv:1007.0296v2 [math.ST].

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for topic models with word embeddings. In *ACL*, pages 795–804.

Xianghua Fu, Ting Wang, Jing Li, Chong Yu, and Wangwang Liu. 2016. Improving distributed word representation and topic model by word-topic mixture model. In *ACML*, pages 190–205.

Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic modeling for short texts with auxiliary word embeddings. In *SIGIR*, pages 165–174.

Ximing Li, Changchun Li, Jinjin Chi, and Jihong Ouyang. 2018a. Short text topic modeling by exploring original documents. *Knowledge and Information Systems*, 56(2):443–462.

Ximing Li, Ang Zhang, Changchun Li, Liantian Guo, Wenting Wang, and Jihong Ouyang. 2018b. Relational biterm topic model: Short-text topic modeling using word embeddings. *The Computer Journal*, 62(3):359–372.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *AAAI*, pages 2418–2424.

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge.

Rishabh Mehrotra, Scott Sanner, Wray Buntine, and Lexing Xie. 2013. Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In *SIGIR*, pages 889–892.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *TACL*, pages 299–313.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543.

Xiaojun Quan, Chunyu Kit, Yong Ge, and Sinno Jialin Pan. 2015. Short and sparse text topic modeling via self-aggregation. In *IJCAI*, pages 2270–2276.

Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K Reddy. 2018. Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In *WWW*, pages 1105–1114.

Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015. Short text clustering via convolutional neural networks. In *NAACL-HLT*, pages 62–69.

Guangxu Xun, Vishravas Gopalakrishnan, Fenglong Ma, Yaliang Li, Jing Gao, and Aidong Zhang. 2016. Topic discovery for short texts using word embeddings. In *ICDM*, pages 1299–1304.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *WWW*, pages 1445–1456. ACM.

Shansong Yang, Weiming Lu, Dezhi Yang, Liang Yao, and Baogang Wei. 2015a. Short text understanding by leveraging knowledge into topic model. In *NAACL*, pages 1252–1237.

Yang Yang, Feifei Wang, Junni Zhang, Jin Xu, and S Yu Philip. 2018. A topic model for co-occurring normal documents and short texts. *World Wide Web*, 21(2):487–513.

Yi Yang, Doug Downey, and Jordan Boyd-Graber. 2015b. Efficient methods for incorporating knowledge into topic models. In *EMNLP*, pages 308–317.

He Zhao, Lan Du, and Wray Buntine. 2017a. Leveraging node attributes for incomplete relational data. In *ICML*, pages 4072–4081.

He Zhao, Lan Du, and Wray Buntine. 2017b. A word embeddings informed focused topic model. In *ACML*, pages 423–438.

He Zhao, Lan Du, and Wray Buntine. 2017c. MetaLDA: A topic model that efficiently incorporates meta information. In *ICDM*, pages 635–644.

He Zhao, Lan Du, Wray Buntine, and Gang Liu. 2018a. Leveraging external information in topic modelling. *Knowledge and Information Systems*, pages 1–33.

He Zhao, Lan Du, Wray Buntine, and Mingyuan Zhou. 2018b. Dirichlet belief networks for topic structure learning. In *NeurIPS*, pages 7966–7977.
Given the probability density functions of Dirichlet and categorical distributions, Eq. (4) can be written as:

$$\Pr(C, -) \propto \prod_d^D \frac{\Gamma(\pi_{d,c_d} + 1) \Gamma(\pi_{d,m} + 1)}{\Gamma(\pi_{d,c_d,m}) \Gamma(\pi_{d,m})} \cdot \prod_m^M \frac{\Gamma(\alpha_c)}{\Gamma(\alpha_c + n_{c,m}^{\text{cluster}})} K \frac{\Gamma(\alpha_k + n_{m,k}^{\text{cluster}})}{\Gamma(\alpha_k)} .$$

(5)

$$\Pr(C, -) \propto \prod_d^D \frac{\pi_{d,c_d} \cdot \pi_{d,m}}{\pi_{d,c_d,m}} .$$

(6)

Given Eq. (6), the conditional probability for Gibbs sampling of $c_d$ can be derived by:

$$\Pr(c_d = m \mid -) \propto \Pr(C^{c_d = m}, W, Z) = \frac{\pi_{d,m} \Gamma(\alpha_c + n_{c,m}^{\text{cluster}})}{\Gamma(\alpha_c + n_{c,m}^{\text{cluster}})} \frac{K \Gamma(\alpha_k + n_{m,k}^{\text{cluster}})}{\Gamma(\alpha_k)} \prod_k^K \frac{\Gamma(\alpha_k + n_{m,k}^{\text{cluster}})}{\Gamma(\alpha_k)} = \pi_{d,m} \frac{\prod_k^K \Gamma(\alpha_k + n_{m,k}^{\text{cluster}})}{\prod_i^I \Gamma(\alpha_c + n_{c,m}^{\text{cluster}})} .$$

(7)

where $n_{d,k}^{\text{doc}} = \sum_i^N_{i} I(z_{d,i} = k)$ and $I(\cdot)$ is the indicator function; $n_{c,m}^{\text{cluster}} = \sum_d^D \sum_m^M I(z_{d,i} = k \& c_d = m)$; $\pi_{d,c} = \sum_m^M \sum_d^D I(z_{d,i} = k \& c_d = m)$; $n_{c,m}^{\text{cluster}} = \sum_k^K n_{m,k}^{\text{cluster}}$; $n_{m,k}^{\text{cluster}} = \sum_d^D n_{d,k}^{\text{cluster}} - n_{d,k}^{\text{doc}}$; $\alpha_c = \sum_k^K \alpha_k$. $\alpha_c$. $\alpha$.

### A.2 Sampling topic assignment $z_{d,i}$:
The sampling of $z_{d,i}$ is similar to the LDA model:

$$\Pr(z_{d,i} = k \mid -) \propto (\alpha_k + n_{c,k}^{\text{cluster}}) \beta_0 + n_{k,v}^{\text{topic}} \beta_0 \ast V + n_{k,v}^{\text{topic}} .$$

(8)

where $n_{v,k}^{\text{topic}} = \sum_d^D \sum_i^N_{i} I(w_{d,i} = v \& z_{d,i} = k)$ and $n_{v,c}^{\text{topic}} = \sum_v^V n_{v,k}^{\text{topic}}$. $n_{v,k}^{\text{topic}}$. $n_{v,c}^{\text{topic}}$.

### A.3 Sampling $\lambda_{l,m}$:
As $\lambda_{l,m}$ is used to construct $\pi_{d,m}$, which is the prior of $\theta_d$, according to Eq. (3), we have:

$$\Pr(\psi_d \mid \pi_d) \propto \frac{\Gamma(\pi_{d,c})}{\Gamma(\pi_{d,c,m} + 1)} \prod_m^M \pi_{d,m} .$$

(9)

According to Zhao et al. (2017c), if we introduce $q_d \sim \text{Beta}(\pi_{d,m}, 1)$, Eq. (9) can be augmented.
as:
\[
\Pr(\pi_d \mid -) \propto \prod_m^{M}(q_d)^{n_{d,m}} \pi_{d,m}. \tag{10}
\]

Recall that \(\pi_{d,m} = \prod_{L}^{L_l=1}(\lambda_{l,m})^{f_{d,l}}\), we can actually extract the terms related to \(\lambda_{l,m}\) to get:
\[
e^{-\lambda_{l,m} \sum_{d : f_{d,l}=1}^{D} \frac{n_{d,m}}{\lambda_{l,m}} \log q_d} \cdot (\lambda_{l,m})^{g_{l,m}}, \tag{11}
\]
where \(g_{l,m} = \sum_{d : f_{d,l}=1}^{D} I_{c_d=1} \cdot (\lambda_{l,m}) \).

Given the above equation, we can sample \(\lambda_{l,m}\) from its gamma posterior:
\[
\lambda_{l,m} \sim \text{Ga}(\mu', \nu'), \quad \mu' = \mu_0 + g_{l,m}, \quad \nu' = \nu_0 - \sum_{L}^{L_l=1} n_{d,m} \log q_d. \tag{12}
\]

### A.4 Sampling \(\lambda'_{l',m}\) for MIGA with hierarchical meta-info:

To incorporate the second-layer labels, \(\lambda'_{l',m}\) can be sampled similarly to \(\lambda_{l,k}\) as follows:
\[
\lambda'_{l',m} \sim \text{Ga}(\mu', \nu'), \quad \mu' = \mu_0 + \sum_{L}^{L_l'=1} x_{l',m}, \quad \nu' = \nu_0 + \sum_{L}^{L_l'=1} \log \left(\frac{y_{l,m} + \mu_0}{\mu_0}\right), \tag{13}
\]
where \(y_{l,m} = \sum_{d : f_{d,l}=1}^{D} \frac{n_{d,m}}{\lambda_{l,m}} \log q_d\) and \(x_{l',m} \sim \text{CRT}(\prod_{L_l'=1}^{L_l} (\lambda'_{l',m})^{f_{d,l',d}} g_{l',m})\). Here, \(h \sim \text{CRT}(n, r)\) stands for the Chinese Restaurant Table distribution (Zhou and Carin, 2015) that generates the number of tables \(h\) seated by \(n\) customers in a Chinese restaurant process with the concentration parameter \(r\) (Buntine and Hutter, 2012).

### A.5 Overall Inference Algorithm for MIGA:

The inference algorithm for MIGA is shown in Algorithm 1.

---

**Algorithm 1** Gibbs sampling algorithm for MIGA  
**Require:** \{\(w_{d,i}\)\}_{d,i}, \{\(f_{d,l}\)\}_{d,l}, \(K, M, \alpha, \beta_0, \mu_0, \text{MaxIteration}\)  
**Ensure:** \{\(c_d\)\}_{d}, \{\(z_{d,i}\)\}_{d,i}, \{\(\lambda_{l,m}\)\}_{l,m}\n
1: Randomly initialise all the latent variables according to the generative process  
2: for \(iter \leftarrow 1\) to \(\text{MaxIteration}\) do  
3: / * Sample the cluster assignments */  
4: for \(d \leftarrow 1\) to \(D\) do  
5: Sample \(c_d\) by Eq. (7)  
6: Update \(n_{\text{cluster}}\)  
7: end for  
8: / * Sample the topics for words */  
9: for \(d \leftarrow 1\) to \(D\) do  
10: for \(i \leftarrow 1\) to \(N_d\) do  
11: For \(w_{d,i} = v\), sample \(z_{d,i}\) by Eq. (8)  
12: Update \(n_{\text{cluster}}, n_{\text{doc}}, n_{\text{topic}}\)  
13: end for  
14: end for  
15: / * Sample \(\lambda_{l,m}\) */  
16: for all \(l,m\) do  
17: Sample \(\lambda_{l,m}\) by Eq. (12)  
18: Recompute \(\pi_{d,m} = \prod_{L}^{L_l=1}(\lambda_{l,m})^{f_{d,l}}\)  
19: end for  
20: end for