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 **M**eta-**I**nfo **G**uided **A**ggregation (**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

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4042–4049
Florence, Italy, July 28 - August 2, 2019.

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 word embeddings from Gaussian distributions; Xun et al. (2016) uses an alternative background model to complement Gaussian topics in GLDA. GPUDMM (Li et al., 2016), GPUPDMM (Li et al., 2017) and SeaNMF (Shi et al., 2018) utilise word semantic relations computed from pre-trained word embeddings. MetaLDA (Zhao et al., 2017c, 2018a), WEIFTM (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.

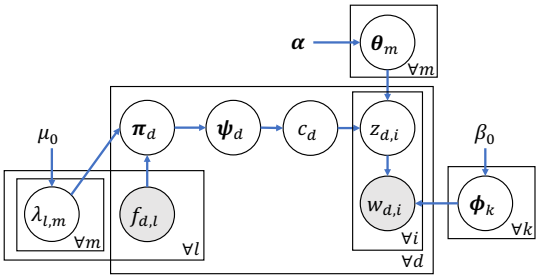

Figure 1: Graphical model of MIGA.

3 The Proposed Model[1]

Given a set of D short documents, the existing self-aggregation methods such as PTM (Zuo et al., 2016a) assume that each document $d \in \{1, \cdots, D\}$ 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} ($i \in \{1, \cdots, N_d\}$) word $w_{d,i}$ in document d with N_d words, we first sample d's cluster assignment $c_d = m \in \{1, \cdots, M\}$ according to its *doc-cluster* distribution, i.e., $\boldsymbol{\psi}_d \in \mathbb{R}_+^M$; and then we sample a topic $z_{d,i} \in \{1, \cdots, K\}$ for word $w_{d,i}$ from the *cluster-topic* distribution $\boldsymbol{\theta}_{c_d} \in \mathbb{R}_+^K$. Given $z_{d,i}$, $w_{d,i}$ is sampled from the *topic-word* distribution $\boldsymbol{\phi}_{z_{d,i}} \in \mathbb{R}_+^V$, 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 $\boldsymbol{\psi}_d$, our model draws it from a document-specific prior $\boldsymbol{\pi}_d \in \mathbb{R}_+^M$, constructed from d's meta-info. Assume that there are L unique labels[2] in a corpus and the labels of document d are encoded in a binary vector $\boldsymbol{f}_d \in \{0, 1\}^L$, where $f_{d,l} = 1$ indicates d has label l. This encoding method allows a

[1]The inference algorithm of the proposed model is elaborated in the appendix.

[2]Hereafter, 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: $\boldsymbol{\theta}_m \sim \text{Dir}_K(\boldsymbol{\alpha})$
2. For each topic k, draw $\boldsymbol{\phi}_k \sim \text{Dir}_V(\beta_0)$
3. For each document d:
   (a) For each cluster m, compute: $\pi_{d,m} = \prod_{l=1}^{L}(\lambda_{l,m})^{f_{d,l}}$
   (b) Draw: $\boldsymbol{\psi}_d \sim \text{Dir}_M(\boldsymbol{\pi}_d)$
   (c) Draw the cluster assignment: $c_d \sim \text{Cat}_M(\boldsymbol{\psi}_d)$
4. For each word i in document d:
   (a) Draw topic: $z_{d,i} \sim \text{Cat}_K(\boldsymbol{\theta}_{c_d})$
   (b) Draw word: $w_{d,i} \sim \text{Cat}_V(\boldsymbol{\phi}_{z_{d,i}})$

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 $\boldsymbol{\psi}_d$, MIGA constructs an informative document-specific Dirichlet prior with parameter π_d computed from the document's labels. Specifically, in Step 3a above, $\lambda_{l,m}$ captures the correlations between label 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 categorised 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 $f'_{l,l'} \in \{0,1\}$ 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:

$$\lambda'_{l',m} \sim \text{Ga}(\mu_0, \mu_0), \quad (1)$$

$$\lambda_{l,m} \sim \text{Ga}\left(\prod_{l'=1}^{L'}(\lambda'_{l',m})^{f'_{l,l'}}, \mu_0\right), \quad (2)$$

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.

Dataset	D	V	avg.N_d	L
Tweets (Mehrotra et al., 2013)	87,638	24,884	11	6
Patents[4]	13,588	3,745	9	$L'=3$ $L=10$
Web Snippets (Li et al., 2016)	12,237	10,052	15	8
Stackoverflow (Xu et al., 2015)	18,287	2,458	5	20
20Newsgroups[5]	10,020	2000	28	$L'=6$ $L=20$

Table 1: Statistics of the datasets. D, V, avg.N_d, and L stand for the number of documents, the vocabulary size, the average document length, and the number of unique labels (L' is the number of unique labels in the second-layer, if available), respectively.

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 $\boldsymbol{\phi}_k \sim \text{Dir}_V(\boldsymbol{\beta}_k)$, where where $\boldsymbol{\beta}_k \in \mathbb{R}_+^V$ 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 $\boldsymbol{\alpha}$; for the

[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.

Dataset	Tweets			Patents			Web Snippets			Stackoverflow			20Newsgroups		
#Clusters	100	200	500	100	200	500	100	200	500	100	200	500	100	200	500
KMeans + TFIDF	-	-	-	0.36	0.44	0.51	0.49	0.60	0.73	0.19	0.24	0.32	0.29	0.31	0.40
KMeans + LDA	0.69	0.70	0.72	0.50	0.52	0.57	0.69	0.71	0.76	0.27	0.30	**0.34**	0.40	0.43	**0.50**
MetaLDA	0.70	0.70	0.64	**0.57**	0.55	0.49	**0.77**	**0.82**	**0.82**	0.31	0.28	0.28	0.39	**0.44**	0.40
GPUDMM	0.59	0.62	0.63	0.36	0.37	0.41	0.69	0.71	0.72	0.17	0.18	0.23	0.25	0.24	0.31
PTM	0.84	0.83	0.84	0.53	0.53	0.55	0.61	0.68	0.77	0.21	0.23	0.26	0.30	0.31	0.34
MIGA	**0.92**	**0.93**	**0.92**	**0.57**	**0.59**	**0.61**	0.71	0.77	0.80	**0.31**	**0.34**	**0.34**	**0.44**	0.43	0.48

(a) Purity

Dataset	Tweets			Patents			Web Snippets			Stackoverflow			20Newsgroups		
#Clusters	100	200	500	100	200	500	100	200	500	100	200	500	100	200	500
KMeans + TFIDF	-	-	-	0.16	0.20	0.23	0.29	0.34	0.39	0.10	0.15	0.23	0.30	0.30	0.34
KMeans + LDA	0.26	0.25	0.25	0.21	0.22	0.25	0.35	0.35	0.36	0.14	0.18	**0.24**	0.30	0.33	**0.38**
MetaLDA	0.29	0.26	0.23	0.25	0.24	0.19	**0.41**	**0.42**	0.41	0.17	0.18	0.21	0.30	0.33	0.32
MetaLDA-eb	0.29	0.26	0.23	0.22	0.23	0.17	0.40	0.38	0.31	0.15	0.14	0.16	0.31	0.32	0.25
GPUDMM	0.21	0.23	0.24	0.10	0.12	0.15	0.36	0.38	0.38	0.06	0.10	0.17	0.15	0.17	0.25
PTM	0.39	0.36	0.35	0.23	0.24	0.25	0.29	0.33	0.38	0.11	0.13	0.16	0.24	0.26	0.28
MIGA	**0.47**	**0.45**	**0.43**	**0.26**	**0.27**	**0.29**	0.35	0.39	0.40	**0.18**	**0.19**	0.22	**0.33**	**0.34**	**0.38**

(b) NMI

Table 2: Purity and NMI for document clustering. The best scores are in boldface.

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)[6], **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]. 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: $\sum_{d=1}^{D}\sum_{i=1}^{N_d}\mathbf{1}_{z_{d,i}=k}$). 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

[6]Original MetaLDA is able to use both document meta-info and word embeddings. Here we used its variant only with document meta-info.

[7]The scores of KMeans + TFIDF on Tweets are not reported because it exceeds the memory of our machine.

[8]We used the Palmetto package (http://palmetto.aksw.org) with a large Wikipedia dump to compute NPMI.

comp
1. use problem using running program
2. fax university internet mail phone
3. thanks help edu advance appreciated
comp.graphics
1. graphics color screen bit mode
2. card drivers driver video windows
3. software support looking products product
comp.sys.ibm.pc.hardware
1. drive disk hard drives floppy
2. mb card controller ide scsi
3. mhz board speed port problem
comp.sys.mac.hardware
1. monitor pc mouse systems box
2. card modem software internal meg
3. like buy looking price new good power want low need
comp.windows.x
1. windows motif application server widget
2. ftp file program package format
3. windows printer font version text

Cluster Num	Documents	Topic
1	purchase store cat outstanding shipping crucial memory upgrades ram sdram ddr service pricing com	computer memory virtual cache apple
	manufacture flash graphics crucial memory upgrades cards usb storage ram media	
	computer selection online memory upgrades ram ddr upgrade specialist	
2	reviews box deal processors shipping computers pricegrabber tax customers prices retail com	digital camera electronics reviews apple
	home store shop manufacturer accessories downloads product online apple ipods	
	reviews forums discussion photography camera articles news digital review	
3	country experience altavista languages search comprehensive web	programming java code language source
	browser documentation hall resources lists faqs apl jhu programming compiler tutorials edu sites java books download	
	introduction programming math textbook downloading on-line edu java	

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 $\lambda'_{l',m}$) 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.

Datasets	Tweets	Patents	Web Snippets	Stack overflow	20News groups
MetaLDA	0.0020	0.0371	0.0501	-0.0767	-0.0146
MetaLDA-eb	0.0062	0.0480	0.0513	-0.0690	-0.0174
GPUDMM	-0.0182	0.0000	0.0368	-0.0992	-0.0535
PTM	**0.0073**	0.0432	0.0408	-0.0675	-0.0282
MIGA	-0.0001	0.0494	0.0544	-0.0430	-0.0178
MIGA-eb	-0.0031	**0.0497**	**0.0786**	**-0.0214**	**-0.0018**

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.

Liangjie Hong and Brian D Davison. 2010. Empirical study of topic modeling in Twitter. In *Workshop on social media analytics*, pages 80–88.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *EACL*, pages 530–539.

Chenliang Li, Yu Duan, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2017. Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems*, 36(2):11.

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, Lantian 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, Vishrawas 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 1232–1237.

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, Wray Buntine, and Gang Liu. 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.

He Zhao, Lan Du, Wray Buntine, and Mingyuan Zhou. 2018c. Inter and intra topic structure learning with word embeddings. In *ICML*, pages 5887–5896.

He Zhao, Piyush Rai, Lan Du, and Wray Buntine. 2018d. Bayesian multi-label learning with sparse features and labels, and label co-occurrences. In *AISTATS*, pages 1943–1951.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *ECIR*, pages 338–349.

M. Zhou and L. Carin. 2015. Negative binomial process count and mixture modeling. *TPAMI*, pages 307–320.

Yuan Zuo, Junjie Wu, Hui Zhang, Hao Lin, Fei Wang, Ke Xu, and Hui Xiong. 2016a. Topic modeling of short texts: A pseudo-document view. In *SIGKDD*, pages 2105–2114.

Yuan Zuo, Jichang Zhao, and Ke Xu. 2016b. Word network topic model: A simple but general solution for short and imbalanced texts. *Knowledge and Information Systems*, 48(2):379–398.

A Appendix on the details of the inference algorithm

We first denote the set of all the words in the target corpus, the set of the topic assignments for all the words, the set of the cluster assignments for all the documents, the set of the labels of all the documents as $\mathcal{W}$, $\mathcal{Z}$, $\mathcal{C}$, and $\mathcal{F}$, respectively. Given the generative process of MIGA, the joint probability of the model is as follows:

$$\Pr(\mathcal{W},\mathcal{Z},\mathcal{C},-\mid\mathcal{F}) = \prod_d^D\prod_i^{N_d}\Pr(z_{d,i}\mid\boldsymbol{\theta_{c_d}})\Pr(w_{d,i}\mid\boldsymbol{\phi_{z_{d,i}}})\cdot \prod_d^D\Pr(c_d\mid\boldsymbol{\psi_d})\Pr(\boldsymbol{\psi_d}\mid\boldsymbol{\pi_d})\cdot\prod_k^K\Pr(\boldsymbol{\phi_k}\mid\beta_0)\cdot \prod_m^M\Pr(\boldsymbol{\theta_m}\mid\boldsymbol{\alpha})\cdot\prod_l^L\prod_m^M\Pr(\lambda_{l,m}). \quad (3)$$

A.1 Sampling cluster assignment c_d:

Now we extract the related terms to the cluster assignments $\mathcal{C}$ in Eq. (3) to have:

$$\Pr(\mathcal{C},-)\propto\prod_d^D\Pr(c_d\mid\boldsymbol{\psi_d})\Pr(\boldsymbol{\psi_d}\mid\boldsymbol{\pi_d})\cdot\prod_d^D\prod_i^{N_d}\Pr(z_{d,i}\mid\boldsymbol{\theta_{c_d}}). \quad (4)$$

Given the probability density functions of Dirichlet and categorical distributions, Eq. (4) can be written as:

$$\Pr(\mathcal{C},-)\propto\prod_d^D\frac{\Gamma(\pi_{d,c_d}+1)\Gamma(\pi_{d,\cdot})}{\Gamma(\pi_{d,c_d})\Gamma(\pi_{d,\cdot}+1)}\cdot \prod_m^M\frac{\Gamma(\alpha_\cdot)}{\Gamma(\alpha_\cdot+n^{\text{cluster}}_{m,\cdot})}\prod_k^K\frac{\Gamma(\alpha_k+n^{\text{cluster}}_{m,k})}{\Gamma(\alpha_k)} \quad (5)$$

$$\propto\prod_d^D\frac{\pi_{d,c_d}}{\pi_{d,\cdot}}\cdot \prod_m^M\frac{\Gamma(\alpha_\cdot)}{\Gamma(\alpha_\cdot+n^{\text{cluster}}_{m,\cdot})}\prod_k^K\frac{\Gamma(\alpha_k+n^{\text{cluster}}_{m,k})}{\Gamma(\alpha_k)}. \quad (6)$$

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

$$\begin{aligned}\Pr(c_d=m\mid-)&\propto\frac{\Pr(\mathcal{C},\mathcal{W},\mathcal{Z})}{\Pr(\mathcal{C}^{\neg d},\mathcal{W}^{\neg d},\mathcal{Z}^{\neg d})}\\ &=\frac{\pi_{d,m}}{\pi_{d,\cdot}}\frac{\Gamma(\alpha_\cdot+n^{\text{cluster}\neg d}_{m,\cdot})}{\Gamma(\alpha_\cdot+n^{\text{cluster}}_{m,\cdot})}\prod_k^K\frac{\Gamma(\alpha_k+n^{\text{cluster}}_{m,k})}{\Gamma(\alpha_k+n^{\text{cluster}\neg d}_{m,k})}\\ &=\frac{\pi_{d,m}}{\pi_{d,\cdot}}\frac{\prod_k^K\left[\prod_j^{n^{\text{doc}}_{d,k}}(\alpha_k+n^{\text{cluster}\neg d}_{m,k}+j-1)\right]}{\prod_i^{N_d}(\alpha_\cdot+n^{\text{cluster}\neg d}_{m,\cdot}+i-1)},\end{aligned} \quad (7)$$

where $n^{\text{doc}}_{d,k}=\sum_i^{N_d}\mathbf{I}_{(z_{d,i}=k)}$ and $\mathbf{I}_{(\cdot)}$ is the indicator function; $n^{\text{cluster}}_{m,k}=\sum_d^D\sum_i^{N_d}\mathbf{I}_{(z_{d,i}=k\&c_d=m)}$; $\pi_{d,\cdot}=\sum_m^M\pi_{d,m}$; $n^{\text{cluster}}_{m,\cdot}=\sum_k^K n^{\text{cluster}}_{m,k}$; $n^{\text{cluster}\neg d}_{m,\cdot}=n^{\text{cluster}}_{m,\cdot}-N_d$; $n^{\text{cluster}\neg d}_{m,k}=n^{\text{cluster}}_{m,k}-n^{\text{doc}}_{d,k}$; $\alpha_\cdot=\sum_k^K\alpha_k$.

A.2 Sampling topic assignment $z_{d,i}$:

The sampling of $z_{d,i}$ is similar to the LDA model:

$$\Pr(z_{d,i}=k\mid-)\propto(\alpha_k+n^{\text{cluster}}_{c_d,k})\frac{\beta_0+n^{\text{topic}}_{k,v}}{\beta_0*V+n^{\text{topic}}_{k,\cdot}}, \quad (8)$$

where $n^{\text{topic}}_{k,v}=\sum_d^D\sum_i^{N_d}\mathbf{I}_{(w_{d,i}=v\&z_{d,i}=k)}$ and $n^{\text{topic}}_{k,\cdot}=\sum_v^V n^{\text{topic}}_{k,v}$.

A.3 Sampling $\lambda_{l,k}$:

As $\lambda_{l,k}$ is used to construct $\boldsymbol{\pi}_d$, which is the prior of $\boldsymbol{\psi}_d$, according to Eq. (3), we have:

$$\Pr(\boldsymbol{\psi}_d\mid\boldsymbol{\pi}_d)\propto\frac{\Gamma(\pi_{d,\cdot})}{\Gamma(\pi_{d,\cdot}+1)}\prod_m^M\pi_{d,m}. \quad (9)$$

According to Zhao et al. (2017c), if we introduce $q_d\sim\text{Beta}(\pi_{d,\cdot},1)$, Eq. (9) can be augmented

as:

$$\Pr(\boldsymbol{\pi}_d \mid -) \propto \prod_{m}^{M} (q_d)^{\pi_{d,m}} \pi_{d,m}. \quad (10)$$

Recall that $\pi_{d,m} = \prod_{l=1}^{L} (\lambda_{l,m})^{f_{d,l}}$, we can actually extract the terms related to $\lambda_{l,m}$ to get:

$$\Pr(\lambda_{l,m} \mid -) \propto e^{-\lambda_{l,m} \sum_{d:f_{d,l}=1}^{D} \frac{\pi_{d,m}}{\lambda_{l,m}} \log \frac{1}{q_d}} \cdot (\lambda_{l,m})^{g_{l,m}}, \quad (11)$$

where $g_{l,m} = \sum_{d}^{D} \mathbf{I}_{f_{d,l}=1 \& c_d=m}$.

Given the above equation, we can sample $\lambda_{l,m}$ from its gamma posterior:

$$\begin{aligned} \lambda_{l,m} &\sim \mathrm{Ga}(\mu, \nu), \\ \mu &= \mu_0 + g_{l,m}, \\ \nu &= \mu_0 - \sum_{d:f_{d,l}=1}^{D} \frac{\pi_{d,m}}{\lambda_{l,m}} \log q_d. \end{aligned} \quad (12)$$

A.4 Sampling $\lambda'_{l',k}$ 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:

$$\begin{aligned} \lambda'_{l',m} &\sim \mathrm{Ga}(\mu', \nu'), \\ \mu' &= \mu_0 + \sum_{l:f'_{l,l'}=1}^{L} x_{l,m}, \\ \nu' &= \mu_0 + \sum_{l:f'_{l,l'}=1}^{L} \log \frac{y_{l,m} + \mu_0}{\mu_0}, \end{aligned} \quad (13)$$

where $y_{l,m} = \sum_{d:f_{d,l}=1}^{D} \frac{\pi_{d,m}}{\lambda_{l,m}} \log \frac{1}{q_d}$ and $x_{l,m} \sim \mathrm{CRT}(\prod_{l'=1}^{L'} (\lambda'_{l',m})^{f'_{l,l'}}, g_{l,m})$. Here, $h \sim \mathrm{CRT}(n, r)$ 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, $\boldsymbol{\alpha}$, β_0, μ_0, $MaxIteration$

Ensure: $\{c_d\}_d$, $\{z_{d,i}\}_{d,i}$, $\{\lambda_{l,m}\}_{l,m}$

```
Randomly initialise all the latent variables according to the generative process
for iter ← 1 to MaxIteration do
    / * Sample the cluster assignments * /
    for d ← 1 to D do
        Sample c_d by Eq. (7)
        Update n^cluster_{m,k}
    end for
    / * Sample the topics for words * /
    for d ← 1 to D do
        for i ← 1 to N_d do
            For w_{d,i} = v, sample z_{d,i} by Eq. (8)
            Update n^cluster_{m,k}, n^doc_{d,k}, n^topic_{k,v}
        end for
    end for
    / * Sample λ_{l,m} * /
    for all l,m do
        Sample λ_{l,m} by Eq. (12)
        Recompute π_{d,m} = ∏_{l=1}^{L} (λ_{l,m})^{f_{d,l}}
    end for
end for
```