A SELF-AGGREGATED HIERARCHICAL TOPIC MODEL FOR SHORT TEXTS

Yue Niu and Hongjie Zhang

University of Science and Technology of China, Hefei, Anhui, China

ABSTRACT

With the growth of the internet, short texts such as tweets from Twitter, news titles from the RSS, or comments from Amazon have become very prevalent. Many tasks need to retrieve information hidden from the content of short texts. So ontology learning methods are proposed for retrieving structured information. Topic hierarchy is a typical ontology that consists of concepts and taxonomy relations between concepts. Current hierarchical topic models are not specially designed for short texts. These methods use word co-occurrence to construct concepts and general-special word relations to construct taxonomy topics. But in short texts, word co-occurrence is sparse and lacking general-special word relations. To overcome this two problems and provide an interpretable result, we designed a hierarchical topic model which aggregates short texts into long documents and constructing topics and relations. Because long documents add additional semantic information, our model can avoid the sparsity of word co-occurrence. In experiments, we measured the quality of concepts by topic coherence metric on four real-world short texts corpus. The result showed that our topic hierarchy is more interpretable than other methods.

KEYWORDS

Hierarchical Topic Model, Texts Analysis, Short Texts, Data Mining.

1. INTRODUCTION

Short texts corpus is a kind of prevalent format of texts on the internet, such as titles, comments, microblogs, questions, etc. Applications are interested in discovering knowledge behind the content of short texts. But it has been recognized a challenging task because that the content is unstructured. To analyze the content, ontology learning methods are proposed [1].

Hierarchical topic model is one kind of typical ontology learning model. This model will construct a tree structure in which nodes are concepts and relations are taxonomy relations between concepts. This hierarchy structure can support a wide range of content analysis tasks. The tree structure can provide a global view so researchers can easily understand what are the current research focus and keywords of this focus. Also, the tree structure can support emergency topic detection so customer service agents can quickly know problems of products from the corpus of user feedback. And for users of microblogs or news, the hierarchical topic tree can help them finding contents of their interests by enhancing search engines.

Currently, researchers propose different kinds of hierarchical topic models, such as nCRP [2], nHDP [3], hPAM [4], etc. These models construct tree structures according to the different hypotheses. But for all these models, short texts sets are not easy to deal with. The reason is that

David C. Wyld et al. (Eds): MLIOB, SIPO, NET, DNLP, SOEA, AISCA - 2021
Pp.149-159, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111212

all these models rely on co-occurrence words and general-special word relations to clustering words into topic. But short texts lack these semantic relations.

Hierarchical model thLDA [5] is proposed to analyze tweets on Twitter. They incorporate word embeddings into a hierarchical model to provide additional semantic information. But all these word embeddings information is generated from Google News, an auxiliary documents corpus. But this auxiliary information may only be suitable for tweets, but unsuitable for other short texts. Word embeddings may be semantic word relations in one domain, but may be non-semantic word relations in other domains. So if the auxiliary corpus is inappropriate, the word embeddings information will bring a lot of non-semantic information into the hierarchical model and will lead to poor performances.

To deal with short texts, another compromised method is proposed at the state of data preparation. This method is called text pooling [6], which draws support from auxiliary information. If short texts have auxiliary information of labels, writers, social relations, etc. This method can incorporate this information by pooling short texts with same labels or writers into long documents. After this data preparation, hierarchical models can deal with these data set as common long documents sets. But this kind of method has some drawbacks. First, the auxiliary information is not very common or easy to acquire. Second, the quality of auxiliary information cannot easily evaluate. Self-aggregated model like PTM [16] generate short texts into long documents to add additional word co-occurrence. But this model cannot provide hierarchical topics.

Inspired by the these methods, our research proposes a self-aggregated hierarchical topic model (shPAM) which aggregates short texts heuristically and does not rely on auxiliary information. We also aggregate short texts into long texts. Long texts can provide a lot of word co-occurrence information and construct additional general-special word relations. So our model can overcome the problems above. We designed long documents as a latent variable. Then we can generate a joint probability distribution with variables: words, topics, long documents, and short texts. Firstly, short texts will implicitly aggregate into latent long documents. Then, we generate topics and topic relations from latent long documents, which are much longer than short texts. In this way, the hierarchical model is generated on latent long documents, avoiding the sparsity of co-occurrence information.

To measure our result, we designed the metric of topic coherence. We adopt PMI score[7] for measuring whether a topic is easy to be interpreted into a comprehensive concept. We compared our model with both baseline models and state-of-the-art models on four different kinds of short texts set. The result showed that the concepts of our model is more coherent compared to other methods.

The remainder of this paper is organized as follows. In section 2, we propose related works. In section 3, we give our model and model inference. We present experimental results in section 4 and conclude our work in section 5.

2. RELATED WORKS

Constructing topic hierarchies is one kind of ontology learning [1]. Ontology learning aims at extracting concepts and relations between concepts from the corpus. Several works only extracting concepts or extracting different kinds of relations. But in our research area, we only extract concepts and taxonomic relations. In this area, methods can be split into two groups. One group methods employ the graphic model which formulates the problem as a joint probability

distribution. Another group method is hierarchical clustering which recursively aggregates layer by layer.

2.1. Topic Model

Different graphic models follow different hypotheses and construct different kinds of hierarchical topic trees. Early proposed methods are hLDA [8] and nCRP [2]. These two methods suggest that one document corresponds to a path in the tree. This suggestion is not very reasonable because one document may correspond to more than one path. Method PAM [9] supposes that only leaf nodes can be translated into concepts. Non-leaf nodes have no explicit meanings. Method hPAM [4] can construct a three-layer topic hierarchy and assume one document may correspond to any topic in the tree. This method can provide only a tree but also a DAG. Method rCRP [10] is very similar to hPAM, but it does not need users to configure the number of topics and the height of the tree. Method nHDP [3] and Ahmed et al[11] have similar models, they suppose that one document corresponds to a sub-tree in the final result. All these methods cannot properly deal with short texts. thLDA [5] is a hierarchical topic model especially for tweet. This model uses word embeddings as the auxiliary information to overcome the sparsity of word co-occurrence. But in this model, if the auxiliary documents are not semantically related with short texts, word embeddings will bring non-semantic information into the model and lead to poor performances.

2.2. Hierarchical Clustering

Hierarchical clustering methods are quite different from the topic models. The graphic model is only a kind of soft-clustering, one document corresponds to several topics with probability. But hierarchical clustering methods cluster documents at every layer. So each document determinately corresponds to one topic at each layer. Bayesian Rose Trees [12] can generate trees with any structure, not only binary trees. Liu et al. [13] construct a topic tree for keywords. This method needs knowledge from auxiliary information. Wang et al.[14] propose a method specialized dealing with content-representative documents such as titles of academic papers. Zhang et al. [15] designed a new combination method specialized for short texts. But their model cannot provide interpretable taxonomic relations as child nodes can be very familiar with father nodes.

2.3. Pooling Method

Pooling methods can help to deal with short texts through auxiliary data. Mehrotra et al. [6] propose a pooling method for tweets by hashtags and can also deal with the corpus that partially has labels. Ahmed et al. [11] also, deal with short texts by pooling them through user id. But auxiliary data are not always available. Besides, pooling through inappropriate auxiliary data may lead to a worse result. PTM [16] is a self-aggregated topic model, but this model cannot generate hierarchical topics.

3. MODEL AND INFERENCE

In this section, we propose our hierarchical model especially for short texts. Firstly, we introduce our model and show how it could supply additional co-occurrence information without auxiliary data. Then, we give the inference method of our model.

3.1. Model

Our model assumed that there are K topics as nodes of the hierarchical tree. The hierarchy has three layers. The first layer is the root topic. The second layer has K_T topics, which are defined as super-topics and root-topic. The third layer has K_t topics as leaf nodes, which are defined as sub-topics and one more topic as super-topic. The total topic number is $K_T + K_t + 1$. For each topic, there is a multinomial distribution over the vocabulary of size V. Also, we assumed that the number of short texts is N_d. The number of words in one short text is represented as N_w. And the number of latent long texts is N_D. There is a multinomial distribution between short tests and latent long texts with parameter φ. So each short text belongs to a long text, and N_D is smaller than N_d. The generation process can be briefly described as follow. Firstly we sample a long text for a special short text. The long text has multinomial distributions over K topics. Then we sample a path of topics in the hierarchy. Having the path, we sample a topic from the path. At last, a word is sampled according to the distribution of the topic.

The generation model is as follows:

1) Sample $\varphi \sim Dir(\alpha)$
2) For each topic z:
   Sample $\eta_z \sim Dir(\gamma)$
3) For each latent long document
   a) For each super-topic:
      Sample $\theta_T \sim Dir(\beta_T)$
   b) For each sub-topic:
      Sample $\theta_t \sim Dir(\beta_t)$
4) For each short document d_i
   a) Sample a latent long document $D \sim Mult(\varphi)$
   b) For each word w_j in short document d_i
      i) Sample a super-topic $z_T \sim Multi(\theta_T^D)$.
         If $z_T = z_{root}$, sample $w_j \sim Multi(\eta_{z_{root}})$
      ii) Else sample a sub-topic $z_t \sim Multi(\theta_t^D)$.
         If $z_t = z_{supe}$, sample $w_j \sim Multi(\eta_{z_T})$
      iii) Else sample $w_j \sim Multi(\eta_{z_t})$

The graphical model can be seen as follows:

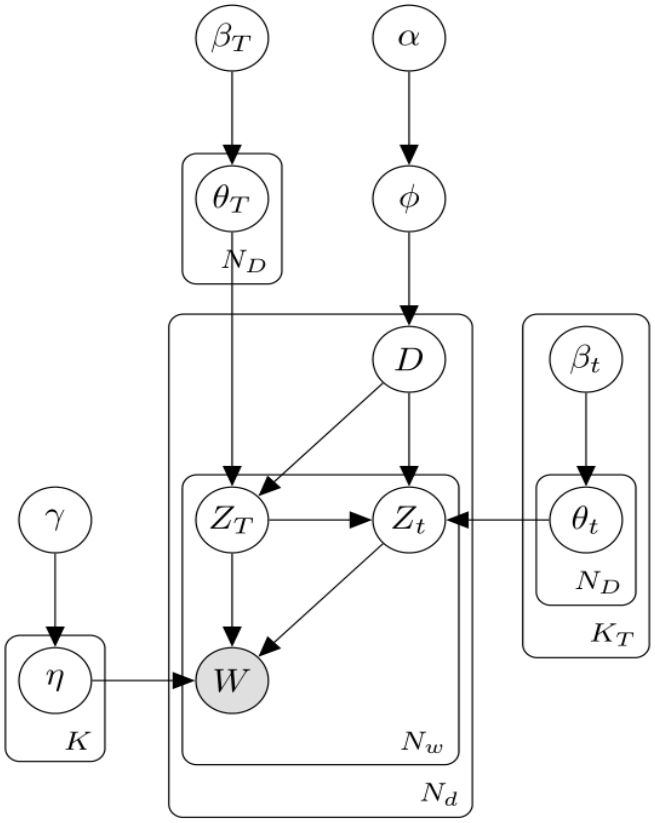

Figure 1. Graphial model of shPAM

The number of latent long documents should be defined carefully. For a short texts corpus, the number of short text N_d is always big. But for each short text, the number of co-occurrence relations M_d between two words are small. Not enough co-occurrence relations will lead to inaccurate results. So in our model, we aggregate short texts by setting $N_D < N_d$. For each long document, the number of co-occurrence relations M_D will be much larger than a single short text, as $M_D > M_d$. Besides, these latent variables are different from latent topics. Latent topics are specific topics with specific concepts. But latent long documents are the aggregations of short texts. A long document can be regarded as a combination of topics. So we should guarantee that $N_D < N_d$ and $N_D > K$. Finally, the graphical model of shPAM can be seen in Figure 1. The solid node W means this variable is all ready known.

In addition, our model proposes a three-layer hierarchical topic model. But the number of layers can be easily extended. We can define sub-sub-topics and recursively processing the topic sampling section in the generation model.

3.2. Inference

As the model is very complex, we train our model by collapsed Gibbs sampling. There are three latent variables for sampling : long document D, super-topic z_T and sub-topic z_t.

Firstly, we sample latent variable D. For every short text m, if it is aggregated into latent long document d, the probability is as follows.

$P(D_i=d|D_{-i},w, z_T, z_t, \alpha, \beta_T, \beta_t) \propto$

$$\frac{n_d+\alpha_d-1}{N+N_d\alpha-1}\prod_{j=1}^{L_m}\frac{1}{n_d+K_{z_T}\beta-J}\prod_{z_T\in m}\prod_{k=1}^{n_m^{z_T}}(n_d^{z_T}+\beta_T-k)\prod_{z_T\in m}\prod_{l=1}^{n_m^{z_T}}\frac{\prod_{z_t\in m}\sum_r^{n_m^{z_T,z_t}}(n_{z_T,d}^{z_t}+\beta_T-r)}{n_d^{z_t}+K\beta_t-l}$$

In this function, we integrate out parameters θ_T, θ_t andφ. n_d is the number of short texts that belong to latent long document d. N is the number of short texts and N_d is the number of latent long documents. L_m is the number of tokens in short text m. n_d is the number of tokens assigned to latent long document d, and K_{z_T} is the number of z_T+1. $n_m^{z_T}$ is the number of tokens belong to short text m with topic z_T, and $n_d^{z_T}$ is the number of tokens belong to latent document d with topic z_T. $n_m^{z_T,z_t}$ is the number of tokens in short text m assigned to z_T and z_t. $n_{z_T,d}^{z_t}$ is the number of tokens of document d assigned to z_T and z_t. $K_{z_t}^{z_T}$ is the number of z_t+1 belongs to a special z_T.

After sampling latent variable D, we sample latent variable z_T and z_t. According to our model, the probability will be decomposed into several conditions. For each word v of short text m which belongs to latent document d. The probability of specified z_T and z_t is as follows:

$P(z_{Ti}=k_T, z_{ti}=k_t|w, D, z_{T,-i}, z_{t,-i}, \gamma, \beta_T, \beta_t)=$

$$\begin{cases} (n_d^{z_T}+\beta_T^{z_T}-1)\dfrac{n_v^{z_T}+\gamma_v-1}{n^{z_T}+V\gamma-1} \\ (n_d^{z_T}+\beta_T^{z_T}-1)\dfrac{n_d^{z_T,z_t}+\beta_t^{z_t}-1}{n_d^{z_T}+K_{z_t}\beta_t^{z_t}-1}\dfrac{n_v^{z_T}+\gamma_v-1}{n^{z_T}+V\gamma-1} \\ (n_d^{z_T}+\beta_T^{z_T}-1)\dfrac{n_d^{z_T,z_t}+\beta_t^{z_t}-1}{n_d^{z_T}+K_{z_t}\beta_t^{z_t}-1}\dfrac{n_v^{z_T,z_t}+\gamma_v-1}{n^{z_T,z_t}+V\gamma-1} \end{cases}$$

Parameters η_z, θ_T, and θ_t are also be integrated out. We calculate $(n_d^{z_T}+\beta_T^{z_T}-1)\frac{n_v^{z_T}+\gamma_v-1}{n^{z_T}+V\gamma-1}$ if word v is assigned to root node. $n_d^{z_T}$ is the number of tokens in latent document d assigned to z_T. $n_v^{z_T}$ is the number of tokens equal to word v and assigned to topic z_T. n^{z_T} is the number of tokens assign to z_T of the document set. We calculate $(n_d^{z_T}+\beta_T^{z_T}-1)\frac{n_d^{z_T,z_t}+\beta_t^{z_t}-1}{n_d^{z_T}+K_{z_t}\beta_t^{z_t}-1}\frac{n_v^{z_T}+\gamma_v-1}{n^{z_T}+V\gamma-1}$ if word w is assigned to a super-topic. $n_d^{z_T}$ is the number of tokens assign to z_T and z_t in document d. We calculate $(n_d^{z_T}+\beta_T^{z_T}-1)\frac{n_d^{z_T,z_t}+\beta_t^{z_t}-1}{n_d^{z_T}+K_{z_t}\beta_t^{z_t}-1}\frac{n_v^{z_T,z_t}+\gamma_v-1}{n^{z_T,z_t}+V\gamma-1}$ if word v is assigned to a sub-topic. $n_v^{z_T,z_t}$ is the number of tokens equal to word v and assigned to topic z_T and z_t. n^{z_T,z_t} is the number of tokens assigned to topic z_T and z_t.

According to the inference equations above, we use Gibbs sampling method to sample the topic tree. Firstly, for each short text m, we sample long document D according to the probability $P(D_i=d|D_{-i},w, z_T, z_t, \alpha, \beta_T, \beta_t)$. If we sample a long document d, then we will calculate the topic tree next step. For each word in m, we firstly sample this word to the root topic or not. If root topic is sampled, we will turn to the next word. If the root topic is sampled, that means this word belongs to child topics. So we sample a topic from the child topics of the root topic. But if no child topic is sampled, that means this word should sample topics from leaf topics. So we sample a topic from leaf topics. All these sampling probabilities are following $P(z_{Ti}=k_T, z_{ti}=k_t|w, D, z_{T,-i}, z_{t,-i}, \gamma, \beta_T, \beta_t)$. The probability of not sampling root topic is 1- $P(z_{Ti}=k_T, z_{ti}=$

$k_t|w, D, z_{T,-i}, z_{t,-i}, \gamma, \beta_T, \beta_t$). And the probability of not sampling child topics is 1-$\sum P(z_{Ti} = k_T, z_{ti} = k_t|w, D, z_{T,-i}, z_{t,-i}, \gamma, \beta_T, \beta_t)$. $\sum P(z_{Ti} = k_T, z_{ti} = k_t|w, D, z_{T,-i}, z_{t,-i}, \gamma, \beta_T, \beta_t)$ is the summary of probabilities of child topics.

4. EXPERIMENTAL RESULTS

In this section, we first introduce the datasets in our experiment and the methods for comparison. Then we describe our evaluation with two metrics.

4.1. Datasets

We adopt four real world datasets to analyze performance. The statistics of these datasets are listed in Table 1. #Documents means the number of short texts. Dictionary size means the number of word. Avg. text size means the average length of short texts.

Table 1. Statistics of datasets.

DataSet	#Documents	Dictionary size	Avg. text size
News Titles	32,603	25,973	18
DBLP	628,363	51,799	7
Tweets	483,552	46,732	6
Reddit	494,704	50,276	7

We minimally pre-processed these datasets by removing stopwords and words that occur only once. We briefly describe them as follows:

News Titles:

This dataset of news titles is collected from RSS feeds of three popular newspaper websites (nyt.com, usatoday.com, reuters.com). There are 32k news titles across 7 categories (Sport, Business, U.S., Health, Sci\&Tech, World, and Entertainment). The description and the titles of news are combined as one short text.

DBLP:

This dataset consists of academic paper titles of computer science. These data are from DBLP, a bibliography website for computer science publications. We obtained 600k paper titles from DBLP database.

Tweets:

This dataset is collected from Twitter website. The content of tweets in Twitter is very informal. So we collected tweets of a specific area. This dataset contains 400k tweets, all about the 2016 United States elections.

Reddit News:

This dataset consists of news titles collected from Reddit. There are 500k news collected from 2008 to 2016.

4.2. Methods & Parameter Settings

In this section, we introduce the methods we implemented for comparison.

hPAM:

We implement this method as the base method. The model of this method suggests that each document has a distribution over the whole tree. The height of the tree is designed by the model. In this experiment, we apply the original model. So the height of the tree is constrained to be three.

nCRP:

This method is the typical model of a tree structure. Its model suggests that each document has a distribution over a path of the tree. Although the height of the tree is decided automatically, we only adopt top three levels of the tree for comparison.

hvHDP:

This method is the state of the art method of DAG structrue model. This model samples every level as an HDP. So this method is proposed under the framework of hHDP. This model not only samples topics at leaf nodes but also sample at non-leaf nodes.

For every method in this experiment, we generate a three-evel tree. For every topic, according to the probability distribution between words and topics, we select top 10 most probable words to represent topic content. Then, we search the PMI score of word co-occurrence between five words. The median PMI score is selected as the representation PMI score of the topic. Last, we calculate the avarage PMI score of the whole tree as the final coherent score.

The parameter setting of our method is as follows. Our method shPAM generates a three level tree with 1 root topic, 5 super-topics, and 10 sub-topics. The hyper-parameter β_T and β_t are all set to be 0.1. Hyper-parameter α is set to be 0.1 and γ is set to be 0.01. The number of latent document is set to be 2000. The parameters of other methods are set according to their papers. Method nCRP and hHDP do not need artificially setting the number of topics of each level. For nCRP, set γ =1.0, m= 100, η= 0.1 and π= 0.5. For hvHDP, set H=0.5, α=10, and λ=1.0. Method hPAM is also set to generate a tree with 5 super-topics and 10 sub-topics. Then set α=0.1, β= 0.01 and γ=10.

4.3. Evaluation Measures

Topic coherence is a common metric to evaluate topics. Here we use PMI score to get topic coherence. This metric needs auxiliary data to calculate PMI score. We chose the latest dump of Wikipedia articles as auxiliary data which contains 5 million documents and 14 million words of vocabulary. Firstly, we build a sliding window of 10 topics. Then we use this sliding window to get word co-occurrence information. Then we calculate PMI score according to the equation: PMI (w1,w2)=$\frac{P(w1,w2)}{P(w1)P(w2)}$. Here, P(w1,w2) means word w1 and word w2 appear in the same sliding window. P(w1) and P(w2) are marginal probabilities. So, according to PMI scores calculated for Wikipedia, we calculate the PMI score of each topics for hierarchical topic models. In our experiment, we choose top 10 words of each topic, and calculate the average PMI score and then calculate the average PMI score of all topics.

4.4. Topic Coherent Evaluation

In this section we show the result of the PMI score and analyse this result.

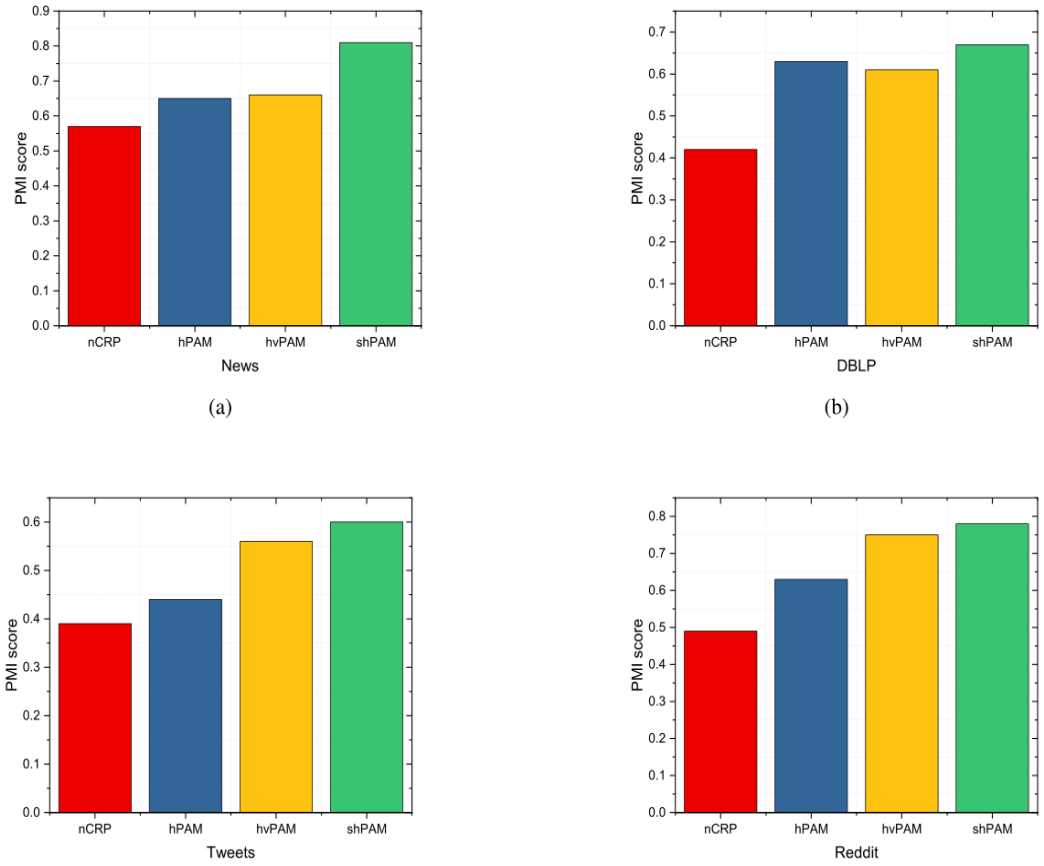

Figure 2. PMI score of 4 datasets

The results of PMI score can be seen in Figure 2. If the topic is more coherent, the PMI score will be higher. Method nCRP has the most un-coherent topics. This model supposes that one document corresponds to a path in the tree. But for short texts, co-occurrence will be more sparsity in subtrees. So the performance of this model is the poorest. Method shPAM, hvHDP, and hPAM all generate a DAG model, which can be seen outperforms tree structured model. Method hPAM is the baseline of DAG model. We can find our method shPAM outperforms it. The state-of-the-art method hvHDP is better than hPAM on dataset News, Tweets, and Reddit. Method hvHDP can provide a more coherent result by automatically changing the number of topics. But these methods cannot provide sufficient word co-occurrence. So the result shows that they all suffer from the sparsity of word co-occurrence. The result of our method shows that by aggregating short texts, we successfully incorporate additional word co-occurrence information into topic model. With more word co-occurrence, our model outperforms the other methods on all datasets.

5. CONCLUSIONS

In this paper, we propose a self-aggregated hierarchical topic model, especially for short texts. Hierarchical models for short texts will suffer from lacking word co-occurrence and general-special word relations. By incorporating long documents as latent variables, our model

aggregates short texts into long documents. Then long documents bring additional word co-occurrence and additional general-special word relations. The experiment on several real-world short texts corpus shows that our model can construct a hierarchy with more coherent topics than the state-of-the-art models. In future works, we will incorporate additional semantic information into hierarchy models such as short texts embeddings. Embeddings information will be helpful to overcome lacking word co-occurrence and general-special word relations.

REFERENCES

[1] Asim, M. N., Wasim, M., Khan, M. U. G., Mahmood, W., & Abbasi, H. M. (2018). A survey of ontology learning techniques and applications. Database, 2018.

[2] Blei, D. M., Griffiths, T. L., & Jordan, M. I. (2010). The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. Journal of the ACM (JACM), 57(2), 1-30.

[3] Paisley, J., Wang, C., Blei, D. M., & Jordan, M. I. (2014). Nested hierarchical Dirichlet processes. IEEE transactions on pattern analysis and machine intelligence, 37(2), 256-270.

[4] Mimno, D., Li, W., & McCallum, A. (2007, June). Mixtures of hierarchical topics with pachinko allocation. In Proceedings of the 24th international conference on Machine learning (pp. 633-640).

[5] Mehrotra, R., Sanner, S., Buntine, W., & Xie, L. (2013, July). Improving lda topic models for microblogs via tweet pooling and automatic labeling. In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (pp. 889-892).

[6] Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In Advances in neural information processing systems (pp. 288-296).

[7] Yu, D., Xu, D., Wang, D., & Ni, Z. (2019). Hierarchical topic modeling of Twitter data for online analytical processing. IEEE Access, 7, 12373-12385.

[8] Blei, D. M., Griffiths, T. L., Jordan, M. I., & Tenenbaum, J. B. (2003, December). Hierarchical topic models and the nested Chinese restaurant process. In NIPS (Vol. 16).

[9] Li, W., & McCallum, A. (2006, June). Pachinko allocation: DAG-structured mixture models of topic correlations. In Proceedings of the 23rd international conference on Machine learning (pp. 577-584).

[10] Kim, J. H., Kim, D., Kim, S., & Oh, A. (2012, October). Modeling topic hierarchies with the recursive chinese restaurant process. In Proceedings of the 21st ACM international conference on Information and knowledge management (pp. 783-792).

[11] Ahmed, A., Hong, L., & Smola, A. (2013, May). Nested chinese restaurant franchise process: Applications to user tracking and document modeling. In International Conference on Machine Learning (pp. 1426-1434). PMLR.

[12] Blundell, C., Teh, Y. W., & Heller, K. A. (2012). Bayesian rose trees. arXiv preprint arXiv:1203.3468.

[13] Song, Y., Liu, S., Liu, X., & Wang, H. (2015). Automatic taxonomy construction from keywords via scalable bayesian rose trees. IEEE Transactions on Knowledge and Data Engineering, 27(7), 1861-1874.

[14] Wang, C., Danilevsky, M., Desai, N., Zhang, Y., Nguyen, P., Taula, T., & Han, J. (2013, August). A phrase mining framework for recursive construction of a topical hierarchy. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 437-445).

[15] Zhang, Y., Mao, W., & Zeng, D. (2015, November). Constructing Topic Hierarchies from Social Media Data. In 2015 IEEE International Conference on Data Mining Workshop (ICDMW) (pp. 1015-1018). IEEE.

[16] Zuo, Y., Li, C., Lin, H., & Wu, J. (2021). Topic Modeling of Short Texts: A Pseudo-Document View with Word Embedding Enhancement. IEEE Transactions on Knowledge and Data Engineering.

AUTHORS

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

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