STTM: A Tool for Short Text Topic Modeling

Jipeng Qiang[1,*], Yun Li[1,*], Yunhao Yuan[1], Wei Liu[1], Xindong Wu[2]

[1]*Department of Computer Science, Yangzhou University, China*

[2]*School of Computing and Informatics, University of Louisiana at Lafayette, USA*

Abstract

Along with the emergence and popularity of social communications on the Internet, topic discovery from short texts becomes fundamental to many applications that require semantic understanding of textual content. As a rising research field, short text topic modeling presents a new and complementary algorithmic methodology to supplement regular text topic modeling, especially targets to limited word co-occurrence information in short texts. This paper presents the first comprehensive open-source package, called STTM, for use in Java that integrates the state-of-the-art models of short text topic modeling algorithms, benchmark datasets, and abundant functions for model inference and evaluation. The package is designed to facilitate the expansion of new methods in this research field and make evaluations between the new approaches and existing ones accessible. STTM is open-sourced at https://github.com/qiang2100/STTM.he

Keywords: Topic Modeling, Short Text, LDA

1. Introduction

Along with the emergence and popularity of social communications (e.g. Twitter and Facebook), short text has become an important information source. Inferring topics from the overwhelming amount of short texts becomes a critical but challenging task for many content analysis tasks [1, 2]. Existing traditional methods for long texts (news reports or papers) such as probabilistic latent semantic analysis (PLSA) [3] and latent Dirichlet allocation (LDA) [4] cannot solve this problem very well since only very limited word co-occurrence information is available in short texts. Therefore, short text topic modeling has already attracted much attention from the machine learning research community in recent years, which aims at overcoming the problem of sparseness in short texts.

[*]Corresponding author

Email addresses: jpqiang@yzu.edu.cn (Jipeng Qiang[1,*]), liyun@yzu.edu.cn (Yun Li[1,*]), yhyuan@yzu.edu.cn (Yunhao Yuan[1]), weiliu@yzu.edu.cn (Wei Liu[1]), xwu@louisiana.edu (Xindong Wu[2])

As a rising research field, short text topic modeling presents a new and complementary algorithmic methodology to supplement topic modeling algorithms for long text, especially targets to limited word co-occurrence information in short texts. To the best of our understanding, there is no comprehensive open-source library available for short text topic modeling algorithms. To facilitate the research on mining latent topics from short texts, we present a novel software package called STTM (Short Text Topic Modeling). The main contribution of STTM lies on three aspects. (1) STTM is the first comprehensive open-source library, which not only includes the state-of-the-art algorithms with a uniform easy-to-use programming interface but also includes a great number of designed modules for the evaluation and application of short text topic modeling algorithms. (2) STTM includes traditional topic modeling algorithms for long texts, which can be conveniently compared with short text topic modeling. (3) STTM is written in Java, easy to use and completely open source. Therefore, new approaches are easily integrated and evaluation through the STTM framework. The package STTM is available at https://github.com/qiang2100/STTM.

2. Design Principles and System Architecture

The framework of STTM follows three basic principles. (1) Preferring integration of existing algorithms rather than implementing them. If the original implementations of algorithms are open, we always attempt to integrate the original codes rather than implement them. The work that we have done is to consolidate the input/output file formats and package these different approaches into some newly designed java classes with a uniform easy-to-use member functions. (2) Including traditional topic modeling algorithms for long texts. The classical topic modeling algorithms (LDA [4] and its variation LF-LDA [5]) are integrated, which is easy for users to comparison of long text topic modeling algorithms and short text topic modeling algorithms. (3) Extendibility. Because short text topic modeling is an emerging research field, many topics such as hierarchical variational models for short texts have not been studied yet. For incorporating future work easily, we try to make the class structures as extendable as possible when designing the core modules of STTM.

Figure 1 shows the hierarchical architecture of STTM. STTM supports the entire knowledge discovery procedure including analysis, inference, evaluation, application for classification and clustering. In the data layer, STTM is able to read a text file, in which each line represents one document. Here, a document is a sequence of words/tokens separated by whitespace characters. If we need to evaluate the algorithm, we also need to read a gold file, in which each line is the class label of one document. In the inference and learning layer, STTM includes a large number of short text topic modeling algorithms. For each model, we not only provide how to train a model on existing corpus but also give how to infer topics on a new/unseen corpus using a pre-trained topic model. For alleviating the problem of sparseness, there are the following three main heuristic strategies in short text topic modeling: window strategy, self-aggregation strategy, and word-embedding strategy. In the application and evaluation layer, STTM presents three aspects about how to evaluate the performance of the algorithms (i.e., topic coherence, clustering, and classification). For topic coherence, we use the point-wise mutual information (PMI) to measure the coherence of topics

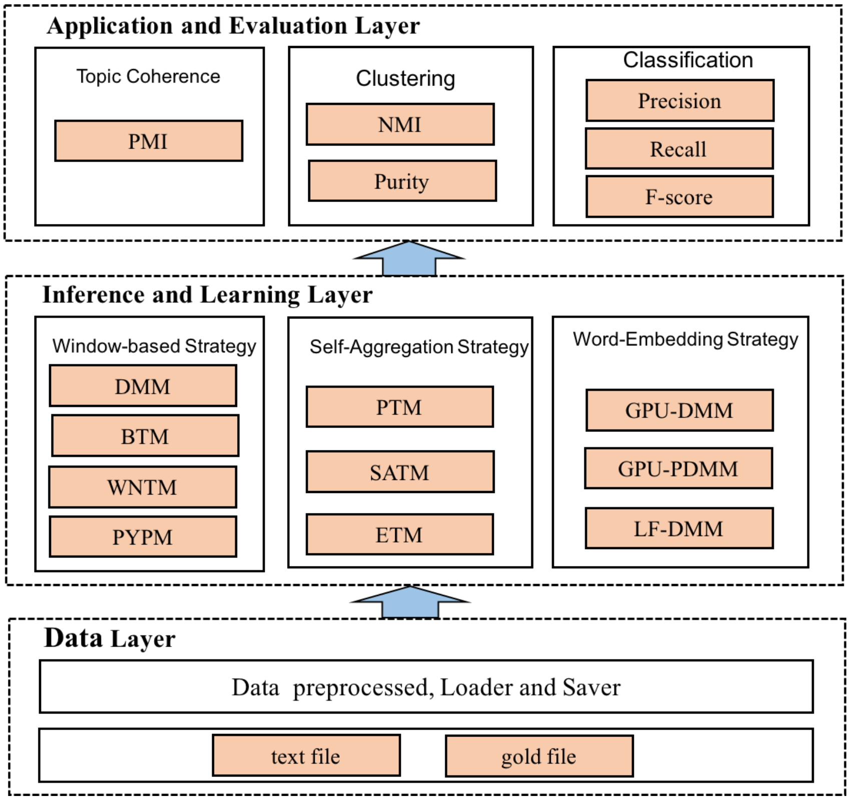


Figure 1: The architecture of STTM

[6]. Short text topic modeling algorithms are widely used for clustering and classification. In the clustering module, STTM provides two measures (NMI and Purity) [7]. Based on the latent semantic representations learned by short text topic modeling, three measures (macro averaged precision, recall and f-score) are used in classifications [1].

3. Algorithms

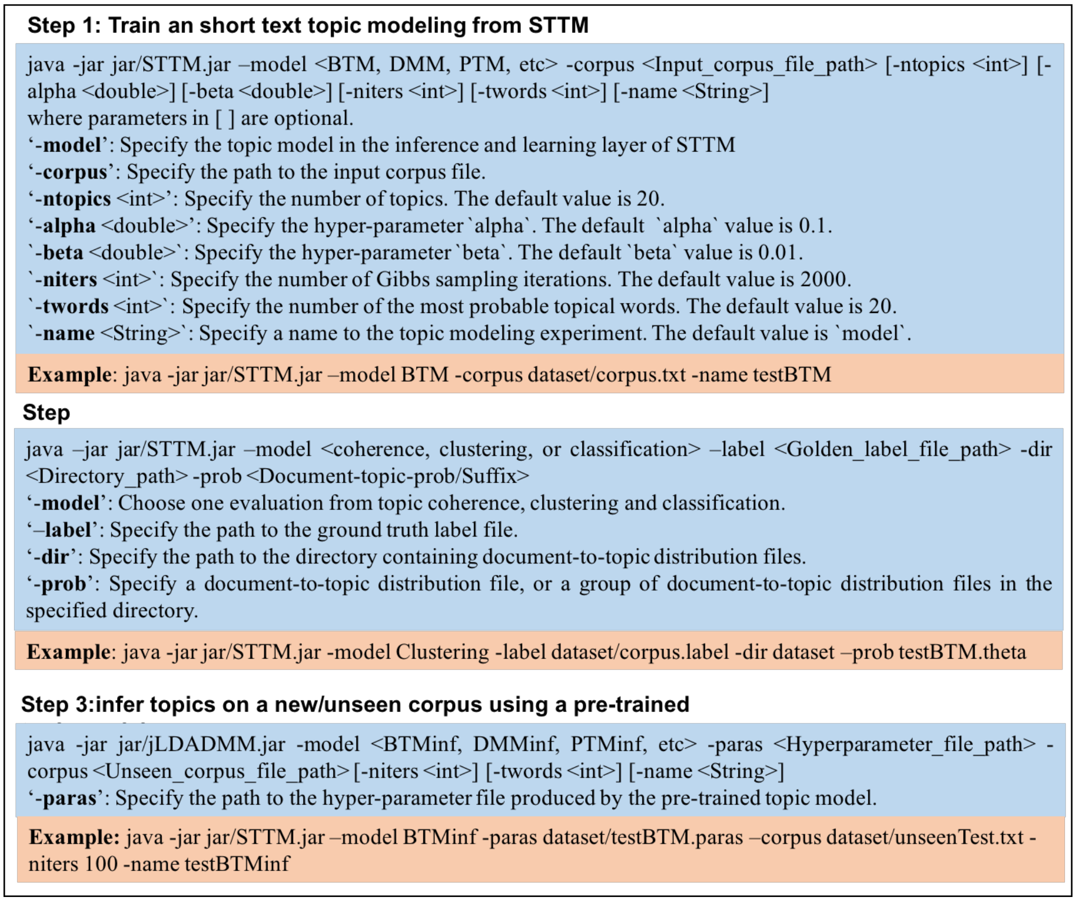
Due to only very limited word co-occurrence information in short texts, how to extract topics from short texts remains a challenging research problem [2]. Three major heuristic strategies have been adopted to deal with how to discover the latent topics from short texts. One follows window-based strategy that two words or all words in one window are sampled from only one latent topic which is totally unsuited to long texts, but it can be suitable for short texts compared to the complex assumption that each text is modeled over a set of topics [7, 8]. Therefore, many models (DMM[9], BTM[1], WNTM[10] and PYPM [11]) for short texts were proposed based on this window-based strategy. The second strategy aggregates short texts into long pseudo-texts before topic inference that can help improve word co-occurrence information. In this framework, STTM based self-aggregation strategy includes three algorithms, PTM [6], SATM [12] and ETM [13]. The last scheme directly leverages recent results by word embeddings that obtain vector representations for words trained on very large corpora to improve the word-topic mapping learned on a smaller corpus. Using word-embedding strategy, STTM integrates the following algorithms, GPU-DMM[14],GPU-PDMM [15] and LF-DMM [5], which are the variations of DMM by incorporating the knowledge of word embeddings.

4. Usage Example

STTM can be easily executed in Linux and Windows systems. The STTM library comes with detailed documentation [1]. Figure 2 gives a sample example about training short text topic modeling from STTM. In this sample example, all models provide a uniform interface function, which can be easily set the parameters for each model.

In the first step, we need to select one algorithm from all short text topic modeling algorithms to learn the latent topics of short texts. Because each algorithm has their self-parameters, STTM provides the options which can be specified by users. Below each step, we give one example about training a model using BTM algorithm. After this step using BTM, the output files including 'testBTM.theta', 'testBTM.phi', ‘testBTM.topWords‘, ‘testBTM.topicAssignments‘ and ‘testDBTM.paras‘ are generated in the ‘dataset‘ folder. In the second step, we can evaluate the performance of the used model in step 1 by choosing one evaluation from topic coherence, clustering and classification. If we choose clustering in this step, the results of NMI and Purity are showed in one file ”testBTM.theta.PurityNMI”. If you need to infer topics for a new/unseen corpus, you can execute the third step.

[1] https://github.com/qiang2100/STTM



Step 1: Train an short text topic modeling from STTM

java -jar jar/STTM.jar –model <BTM, DMM, PTM, etc> -corpus <Input_corpus_file_path> [-ntopics <int>] [-alpha <double>] [-beta <double>] [-niters <int>] [-twords <int>] [-name <String>]
where parameters in [ ] are optional.
‘**-model**’: Specify the topic model in the inference and learning layer of STTM
‘**-corpus**’: Specify the path to the input corpus file.
‘**-ntopics** <int>’: Specify the number of topics. The default value is 20.
‘**-alpha** <double>’: Specify the hyper-parameter `alpha`. The default `alpha` value is 0.1.
`**-beta** <double>`: Specify the hyper-parameter `beta`. The default `beta` value is 0.01.
`**-niters** <int>`: Specify the number of Gibbs sampling iterations. The default value is 2000.
`**-twords** <int>`: Specify the number of the most probable topical words. The default value is 20.
`**-name** <String>`: Specify a name to the topic modeling experiment. The default value is `model`.

Example: java -jar jar/STTM.jar –model BTM -corpus dataset/corpus.txt -name testBTM

Step

java –jar jar/STTM.jar –model <coherence, clustering, or classification> –label <Golden_label_file_path> -dir <Directory_path> -prob <Document-topic-prob/Suffix>
‘**-model**’: Choose one evaluation from topic coherence, clustering and classification.
‘**–label**’: Specify the path to the ground truth label file.
‘**-dir**’: Specify the path to the directory containing document-to-topic distribution files.
‘**-prob**’: Specify a document-to-topic distribution file, or a group of document-to-topic distribution files in the specified directory.

Example: java -jar jar/STTM.jar -model Clustering -label dataset/corpus.label -dir dataset –prob testBTM.theta

Step 3:infer topics on a new/unseen corpus using a pre-trained

java -jar jar/jLDADMM.jar -model <BTMinf, DMMinf, PTMinf, etc> -paras <Hyperparameter_file_path> -corpus <Unseen_corpus_file_path> [-niters <int>] [-twords <int>] [-name <String>]
‘**-paras**’: Specify the path to the hyper-parameter file produced by the pre-trained topic model.

Example: java -jar jar/STTM.jar –model BTMinf -paras dataset/testBTM.paras –corpus dataset/unseenTest.txt -niters 100 -name testBTMinf

Figure 2: A sample example for a basic usage

5. Conclusion and Future Work

STTM is an easy-to-use open-source library for short text topic modeling to facilitate research efforts in machine learning and data mining. The recent version of STTM is consisted of many short text topic modeling algorithms, unseen corpus inferring and beneficial modules supporting different evaluations. Through the STTM framework, we hope that programmers or researchers can easily use short text topic modeling algorithms to discover latent topics fro short texts, and new methods are easily integrated into STTM framework and evaluation.

Acknowledgement

This research is partially supported by the National Natural Science Foundation of China under grants (61703362, 61702441, 61402203), theNatural Science Foundation of Jiangsu Province of China under grants (BK20170513, BK20161338), the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province of China under grant 17KJB520045, and the Science and Technology Planning Project of Yangzhou of China under grant YZ2016238.

References

References

[1] X. Cheng, X. Yan, Y. Lan, J. Guo, Btm: Topic modeling over short texts, Knowledge and Data Engineering, IEEE Transactions on 26 (12) (2014) 2928–2941.

[2] X. Wang, Y. Wang, W. Zuo, G. Cai, Exploring social context for topic identification in short and noisy texts, in: Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[3] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1999, pp. 50–57.

[4] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, the Journal of machine Learning research 3 (2003) 993–1022.

[5] D. Q. Nguyen, R. Billingsley, L. Du, M. Johnson, Improving topic models with latent feature word representations, Transactions of the Association for Computational Linguistics 3 (2015) 299–313.

[6] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu, H. Xiong, Topic modeling of short texts: A pseudo-document view, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, ACM, 2016, pp. 2105–2114.

[7] X. Yan, J. Guo, Y. Lan, J. Xu, X. Cheng, A probabilistic model for bursty topic discovery in microblogs., in: AAAI, 2015, pp. 353–359.

[8] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, X. Li, Comparing twitter and traditional media using topic models, in: Advances in Information Retrieval, 2011, pp. 338–349.

[9] J. Yin, J. Wang, A dirichlet multinomial mixture model-based approach for short text clustering, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 233–242.

[10] Y. Zuo, J. Zhao, K. Xu, Word network topic model: a simple but general solution for short and imbalanced texts, Knowledge and Information Systems 48 (2) (2016) 379–398.

[11] J. Qiang, Y. Li, Y. Yuan, X. Wu, Short text clustering based on pitman-yor process mixture model, Applied Intelligence 48 (7) (2018) 1802–1812.

[12] X. Quan, C. Kit, Y. Ge, S. J. Pan, Short and sparse text topic modeling via self-aggregation, in: Proceedings of the 24th International Conference on Artificial Intelligence, 2015, pp. 2270–2276.

[13] J. Qiang, P. Chen, T. Wang, X. Wu, Topic modeling over short texts by incorporating word embeddings, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2017, pp. 363–374.

[14] C. Li, H. Wang, Z. Zhang, A. Sun, Z. Ma, Topic modeling for short texts with auxiliary word embeddings, in: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, ACM, 2016, pp. 165–174.

[15] C. Li, Y. Duan, H. Wang, Z. Zhang, A. Sun, Z. Ma, Enhancing topic modeling for short texts with auxiliary word embeddings, ACM Transactions on Information Systems (TOIS) 36 (2) (2017) 11.