International Journal of Semantic Computing
Vol. 14, No. 2 (2020) 273–293

DOI: 10.1142/S1793351X20400103

World Scientific
www.worldscientific.com

Transitive Topic Modeling with Conversational Structure Context: Discovering Topics that are Most Popular in Online Discussions

Yingcheng Sun

Department of Computer and Data Sciences
Case Western Reserve University
Cleveland, Ohio 44106, USA
yxs489@case.edu

Richard Kolacinski* and Kenneth Loparo†

Department of Electrical, Computer, and Systems Engineering
Case Western Reserve University
Cleveland, Ohio 44106, USA
**rmk4@case.edu*
†*kal4@case.edu*

With the explosive growth of online discussions published everyday on social media platforms, comprehension and discovery of the most popular topics have become a challenging problem. Conventional topic models have had limited success in online discussions because the corpus is extremely sparse and noisy. To overcome their limitations, we use the discussion thread tree structure and propose a "popularity" metric to quantify the number of replies to a comment to extend the frequency of word occurrences, and the "transitivity" concept to characterize topic dependency among nodes in a nested discussion thread. We build a Conversational Structure Aware Topic Model (CSATM) based on popularity and transitivity to infer topics and their assignments to comments. Experiments on real forum datasets are used to demonstrate improved performance for topic extraction with six different measurements of coherence and impressive accuracy for topic assignments.

Keywords: Online discussions; topic modeling; conversational structure.

1. Introduction

With the prevalence of content sharing platforms, such as online forums, microblogs, social networks, photo and video sharing websites, more and more people like to express and share their opinions on the Internet. Modern news websites provide commenting facilities for their readers to freely post and reply. The increasing popularity of such platforms results in huge amounts of online discussions each day and raises a question: what topics are most popular and highly discussed? Automatically modeling topics from massive texts can help people better understand the main clues and semantic structures, and can also be useful to downstream

applications such as discussion summarization [1], stance detection [2], event tracking [3], and so on.

"Topic" is a certain distribution of words in a document and "Topic model" is a type of statistical model for discovering the abstract semantic structure "topics" that occur in a collection of documents. Conventional topic models, like probabilistic Latent Semantic Analysis (pLSA) [4] and Latent Dirichlet Allocation (LDA) [5] assume documents have topics that can be inferred from word–document co-occurrences. They have achieved great success in modeling long text documents over the past decades, but may not work well when directly applied to short texts that dominate online discussions for two reasons about the data: (1) **Sparse**: The occurrences of words in short documents have a diminished discriminative role compared to lengthy documents where the model has sufficient word counts to determine how words are related. [6] (2) **Noisy**: Comment threads often contain unproductive banter, insults, and cursing, with users often "shouting" over each other [7], and people sometimes publish "unserious" response posts that are unrelated to the discussion topics [8]. Noisy comments perhaps could be used for sentiment analysis, but are significant disturbances when extracting topics from discussion threads.

Based on results from the existing literature, there is a need for additional work that specifically addresses problems with using short text segments for topic modeling. In this paper, we use the tree structure that each discussion thread inherently exhibits based on the relationship between postings and replies to enrich the background information of each comment. Figure 1 illustrates a typical discussion thread of user comments on a submitted question and its corresponding tree structure.

In Fig. 1, the occurrence frequency of each word in the possible topic "concept of 'how all roads work' completely blows your mind" equals to or even less than those "non-topical" words, making it very difficult to be modeled using conventional topic models. The word distribution is closed to uniform, and the words with higher frequency are not even topic words. However, we can see that different comment nodes have different numbers of replies, and nodes (nodes 0 and 1) leading the topics have more replies than others, and those nodes are also in relatively "higher" positions in the discussion tree, above their topic "following" nodes. Motivated by this observation, we propose "popularity" metric to measure the number of replies to a comment as an extension to the frequency of word occurrence. We also observe that the topic distribution of a node is dependent on its parent because comments in reply to the content of their parents form a conversational thread. We use this "transitivity" characteristic as context information to reduce the inaccuracy of topic assignments to comments, especially for those "noisy" ones, like comment 9 in Fig. 1. Based on the above two characteristics, we build a Conversational Structure Aware Topic Model (CSATM) that makes the topics modeled meaningful and usable, and robust to noisy comments.

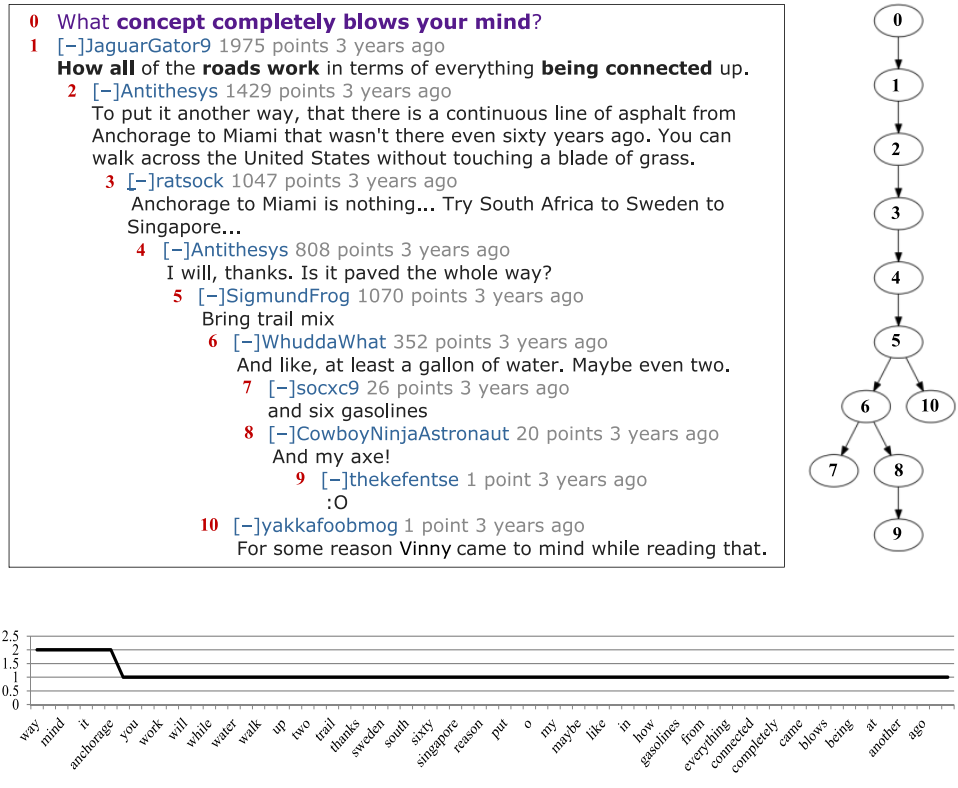

Fig. 1. An example thread of user comments on the posted question: "the concept completely blows your mind" [a] with the original nested discussion on the left and its corresponding Tree structure on the right. i: the ith comment. The figure on the bottom shows the word distribution that is closed to uniform.

The rest of this paper is organized as follows. In Sec. 2, we discuss the related work and somewhat similar approaches. In Sec. 3, we propose the CSATM model and describe the inference method for the model. In Sec. 4, we introduce the datasets, comparison models and evaluation metrics, as well as the experimental results. In Sec. 5, we analyze the application of CSATM to a specific example. We conclude our work in Sec. 6.

2. Related Research

Topic models aim to discover latent semantic information, i.e. topics, from texts and have been extensively studied. LDA [5] is a widely used topic model that represents a document as a mixture of latent topics to be inferred, where a topic is modeled as a multinomial distribution of words. Nevertheless, prior research has demonstrated that topic models only focusing on word–document co-occurrences are not suitable

[a] https://www.reddit.com/r/AskReddit/comments/3dtyke/what_concept_completely_blows_your_mind.

for short and informal texts like Tweets, reviews, and online comments due to data sparsity and noise [9]. Therefore, three main strategies are proposed by recent researchers to tackle these problems and we provide a brief overview of them.

2.1. *Merging shorts texts into long pseudo documents*

The idea of this strategy is merging related short texts together and applying standard topic modeling techniques on the pooled documents. Auxiliary contextual information is used during the merging process, like authors, time, locations, hashtags, conversations, etc. For example, Weng *et al.* [10], Hong and Davison [6], and Zhao *et al.* [11] heuristically aggregate messages posted by the same user or that share the same words before conventional topic models are applied. Alvarez-Melis and Saveski [12] group tweets together occurring in the same user-to-user conversation. Ramage *et al.* [13] and Mehrotra *et al.* [14] employ hashtags as labels to train supervised topic models. The performance of these models can be compromised when facing unseen topics that are irrelevant to any hashtag in the training data.

In practice, auxiliary information is not always available or just too costly for deployment, so models without using auxiliary information have been put forward, like Self-Aggregation-based Topic Model (SATM) [15], Pseudo-document-based Topic Model (PTM) [16], etc. However, those models still could not deal with the case when the data is extremely sparse and noisy like the example Fig. 1 shows, and no prior knowledge is given to ensure the quality of text aggregation, that will further affect the performance of topic inference.

2.2. *Building internal relationships of words*

This strategy uses the internal semantic relationships of words to overcome the problem of lacking word co-occurrence, and the semantic information of words has been effectively captured by deep-neural network-based word embedding techniques. Several attempts [17, 18] have been made to discover topics for short texts by leveraging semantic information of words from existing sources. These topic models rely on a meaningful embedding of words obtained through training on a large-scale high-quality external corpus, which should be both in the same domain and language as the data used for topic modeling.

The SeaNMF [9] model learns the semantic relationship between words and their context from a skip-gram view of the corpus. The Biterm Topic Model (BTM) [19] and the RNN-IDF-based Biterm Short-text Topic Model (RIBSTM) [20] model biterm co-occurrences in the entire corpus to enhance topic discovery. Latent Feature LDA (LFTM) [21] incorporates latent feature vector representations of words. The relational BTM model (R-BTM) [22], links short texts using a similarity list of words computed using an embedding of the words. However, such external resources are not always available [23–25], and the word relationships are not enough to overcome the noisy corpus issue when building topic models [26–29].

2.3. *Leveraging discussion tree structure as prior*

The third line of research focuses on enriching prior knowledge when training the topic model. LeadLDA [30] distinguishes reply nodes into "leaders" and "followers" in the conversation tree, and models the distribution of topical and non-topical words from "leaders" and "followers", respectively. To detect "leaders" and "followers" in the tree structure, the first step is to extract all root-to-leaf paths and then classifying nodes in each path using a supervised learning model after labeling, and then combing all paths [31]. Extracting and combing paths is time consuming and labeling is labor intensive, so LeadLDA may not be suitable for large online discussion datasets. Li *et al.* [32] exploits discourse in conversations and joins conversational discourse and latent topics together for topic modeling. This model also organizes microblog posts as a conversation tree structure, but does not consider topic hierarchies and model robustness issue like our proposed model [33].

Hierarchical Dirichlet Process (HDP) [34] and Nested Hierarchical Dirichlet Process (nHDP) [35] can build hierarchical topic models with nonparametric Bayesian networks, but they model the hierarchical structure of topics, not the documents. In online discussions, if we treat each comment as a document, the comment it replies to and its following replies all provide plentiful clues for its topic inference, which is not discussed in HDP or nHDP. To overcome the text sparsity and noisy issues [36–38], learning-based methods are also explored [39–41].

In this paper, we will introduce a model that uses the conversational structure of a discussion thread inherently has to improve the topic modeling performance for short texts within online discussions.

3. Conversational Structure Aware Topic Model

Our model extends the LDA model by adding the structural relationships among nodes in a discussion tree as context information for each online comment. With the conversational structure, we observe the "popularity" and "transitivity" characteristics of topics in online discussions. We will introduce the intuitions on "popularity" and "transitivity" and how we use them in our model to make extracted topics meaningful and usable.

3.1. *Topic popularity*

"Topic" is the subject of a discourse or of a section of a discourse [42], so it needs to be discussed and popular. In online discussions, users can easily participate by submitting comments or writing replies to those that draw their attention. In writing a reply, a user reads the initial post or headline, browses the comments and selects one for a reply. By writing a reply, a user explicitly expresses their interest in the topic(s) in the discussion thread, thereby increasing their popularity and enlarging the discussion tree by adding leaf nodes. The main topics of a reply may not be closely related to comments located at a distance in the discussion thread, but will definitely

be responsive to the comment it is directly replying to. We thus proposed the "popularity" intuition:

(1) **The popularity of topics discussed in a comment node is positively related to the number of replies.**

For intuition (1), the word "popularity" is commonly used as the state or condition of a person or item being liked by the people. The popularity of an item usually depends on the number of people that support it. As the readers to a book and the audience to a movie, the popularity of a topic can be measured by the number of people that are involved in its discussions. There may be various reasons that a topic becomes popular like its creation time, the celebrity of its author or the topic itself, but the reasons are not what we are going to discuss in this chapter. We are more interested in finding the most popular and influential topics in an online discussion thread, and we also believe that such kind of topics should be extracted by topic models. As the discussion tree example Fig. 2(a) shows, root node 1 may put forward a main topic with three replies: nodes 2, 3, and 4. If we assume these three nodes discuss three "sub-topics", then the sub-topic in node 3 is the most popular because it receives the most responses and should be assigned with higher possibility.

Following intuition (1), the "popularity" p_i of node i depends on all replies in its subtrees, and replies in different level have different weights but the same weight in the same level; so p_i can be written as

$$p_i = \sum_{i} \sum_{n_l} w_l * \text{node} = \sum_{d_i} w_l * p_j, \tag{1}$$

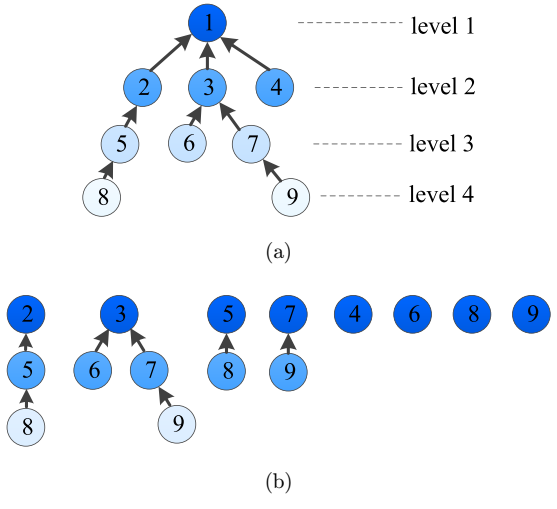

Fig. 2. (Color online) (a) Example of a discussion tree with four levels. (b) Subtrees used for calculating popularity scores of nodes 2 to 9. Shade of color represents topic "influence" of the root, the deeper the stronger the influence.

where n_l is the number of nodes in level l, and w_l is the weight for nodes in level l. We can also write the popularity score of a node as the sum of its children's popularity scores by iterative accumulation, and d_i is the degree of node i. We need to be careful that all counts should be taken in node i's subtree. As Fig. 2(b) shows, node 2's popularity is calculated only on nodes 5 and 8, not on any other node. Also, we set the initial popularity of any node as 1 in this section, so the popularity value of a node without any reply is 1 that is its initial value, like nodes 4, 6, 8 and 9 in Fig. 2. For nodes with replies, like nodes 1, 2, 3, 5 and 7, their popularity values are the sum of the initial popularity and the popularity of replies. According to intuition (2), the popularity of replies in different levels does not have the same weight.

3.2. *Topic transitivity*

In a discussion tree, there are two basic structures: tree style and chain style. With the topic popularity intuition, more replies to a comment, more popular it is, and more possible it generates the topic. Given the examples of tree style and chain style structures shown in Fig. 3, do the top nodes have the same popularity?

It is found that 64–72% of all comments are shifted from their original topics [43], and that topic shift [44] or the topic drift [45] phenomenon make the transitivity process with some "loss", so the "topic influence" of a root decreases when the discussion thread gets longer. Therefore, we assume that the top node in the tree style structure example will have higher popularity than the one in the chain style structure example, and propose the topic transitivity intuition:

(2) **The topic distribution of a node is dependent on its ancestors, and the dependency is negatively related to the distance from the node to its ancestor.**

For intuition (2), let's assume there is a comment node i in the discussion tree t. Users can choose any comment to reply in t, but if i is chosen, it indicates that the topics in comment node i attract the users more than other nodes. The newly added child node to i continue the topics discussed in i, making topic transitive from i to its

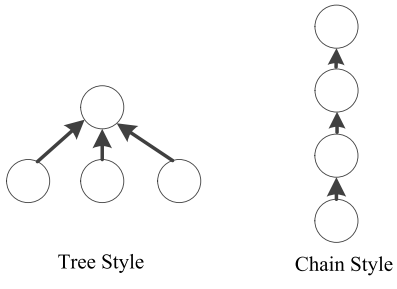

Fig. 3. Examples of two basic structures of a discussion tree: tree style structure and chain style structure.

children, but with a certain probability to discuss other topics. As the example shown in Fig. 2(a), the topic introduced in node 1 spreads across the entire tree, but its influence will weaken from level 1 to level 4 because of the topic transitivity loss. We thus use a decreasing sequence to model the weight w_l in Eq. (1) and we assume that nodes in level l of the subtree have the same weight. We list three different options as the decreasing sequence:

(a) Arithmetic progression

$$w_{al} = c - (l-1)d;$$

(b) Geometric progression

$$w_{gl} = cr^{l-1};$$

(c) Harmonic progression with "gravity" power

$$w_{hl} = (c + (l-1)b)^{-G},$$

where c is a constant, d is the common difference for arithmetic progression, l is the number of the level, and r is the common ratio for the geometric sequence. G is the "gravity" power controlling the fall rate of weights for harmonic progression, and the weight decreases faster the larger G is. If $G = 1$, it becomes general harmonic series, where c and b are real numbers. From arithmetic progression to harmonic progression, the weigh distribution curve will become smoother. Figure 4 shows their differences.

The distribution of popularity score computed by arithmetic progression is sharper, meaning that nodes leading a discussion with a large number of descendants will be given more weights than the other two, so if the dataset is very sparse or topical words are corrupted by noises, the arithmetic progression will be a better choice. From arithmetic progression to harmonic progression, the weight distribution

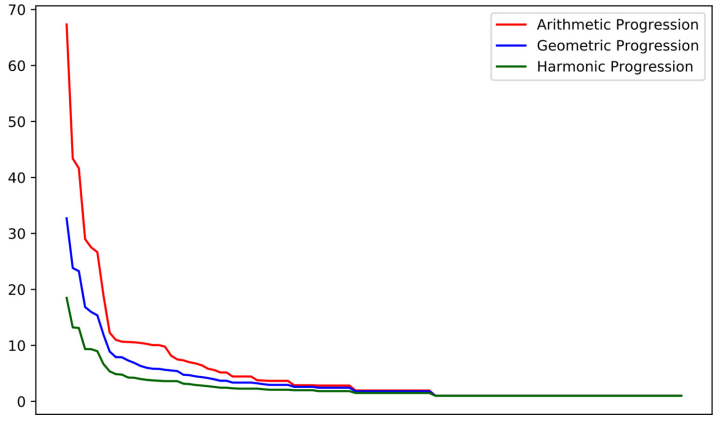

Fig. 4. Distributions of popularity scores calculated by arithmetic, geometric and harmonic progressions on the same datasets.

curve becomes smoother and smoother. The choice of sequence is based on the word distribution of datasets, and other sequence can also be used if it fits the modeling requirements.

3.3. *Model inference*

CSATM extends the LDA model by integrating the popularity property for each online comment. The latent variables of interest are the topic assignments for word tokens z, the comment level topic distribution θ and the topic — word distribution ϕ. The multinomial distribution θ and ϕ can be efficiently marginalized due to the conjugate Dirichlet-multinomial design, we thus only need to sample the topic assignments z. It is computationally intractable to compute the exact posterior distribution using Gibbs sampling for approximating inference. To perform Gibbs sampling, we first choose initial states for the Markov chain randomly. Then we calculate the conditional distribution $p(z_i = k|z^{-i}, \boldsymbol{w}, \boldsymbol{p_c}, \alpha, \beta)$ for each word, where the superscript '$-i$' signifies leaving the ith token out of the calculation, $\boldsymbol{w}$ is the global word set, and $\boldsymbol{p_c}$ is the popularity score for comment c. By applying the chain rule on the joint probability of the data, we can obtain the conditional probability as

$$p(z_i = k|z^{-i}, \boldsymbol{w}, \boldsymbol{p_c}, \alpha, \beta) \propto (n_{k,c}^{-i}\lambda p_c + \alpha_k)\frac{n_{k,w}^{-i}\lambda p_c + \beta_w}{\sum_w n_{k,w}^{-i}\lambda p_c + \beta_w},$$

where $n_{k,c}$ is the number of words in comment c that are assigned to topic k, and $n_{k,w}$ is the number of times that topic k is assigned to word term w, both of which are scaled by the popularity score, and λ is the scaling ratio. Following the conventions of LDA, here we use symmetric Dirichlet priors α and β. Based on the topic assignments of word occurrences, we can estimate the topic-word distributions ϕ and global topic distributions θ as

$$\phi_{k,w} = \frac{\beta_w + n_{k,w}\lambda p_c}{\beta_w + \sum_w n_{k,w}\lambda p_c};$$

$$\theta_{k,c} = \frac{\alpha_w + n_{k,c}\lambda p_c}{\alpha_w + \sum_k n_{k,c}\lambda p_c}.$$

3.4. *Topic assignment*

After discovering usable topics from the corpus, we want the correspondence of topic assignments to documents to be meaningful. Conventional topic assignment methods do not consider the document context information, because for most of the corpus, documents are not dependent. However, comments in online discussions demonstrate clear topic dependency through their nested reply relationships, so we propose a new topic assignment strategy. With CSATM, we obtain the topic distribution for each given comment, and then work out new topic assignments for the comments using

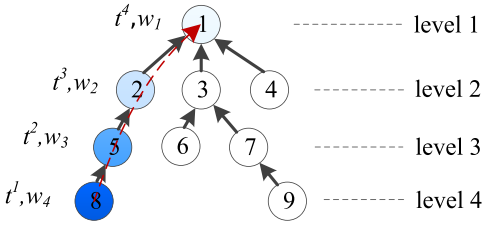

Fig. 5. (Color online) Topic assignment using the topic "transitivity" property in a discussion tree, determining the topic distribution of node 8. The shades of color represent topic dependency, the deeper the color the greater the dependency, with white representing no dependency.

the topic transitivity property

$$t_i' = \frac{\sum_{j=1}^{l_i} w_{l_i-j+1} t_i^j}{\sum_{j=1}^{l_i} w_{l_i-j+1}}, \quad i = 1, \ldots, N,$$

where t_i' is the new topic assignment compared to the original assignment t_i^j for comment i, and j is the relative order in the path from comment node i to the root, and l_i is the level where node i is located, and w is the weight of level l_i used for calculating the popularity score.

In Fig. 5, the topic distribution of node 8 depends on that of nodes in its path to the root, which are nodes 5, 2 and 1, and does not depend on any node out of the path to the root in terms of the topic distribution. The dependency weakens as the level increases because comments indicate stronger interests in their parent nodes they reply to in upper level than nodes in other levels as discussed intuition (2). By using this new strategy, we can reduce the inaccuracy and uncertainty when assigning topics to noisy comments.

4. Experiment

In this section, we evaluate the proposed CSATM against LDA and several state-of-the-art baseline methods on two real world datasets. We report the performance in terms of six different coherence measures, and compare the accuracy for topic assignments.

4.1. *Datasets, compared models, and parameter settings*

In the experiment, we use the Reddit dataset. Reddit is an online discussion website.[b] Registered members can submit content to the site such as links, text posts, or images, and write comments or reply other comments. Posts are organized by subject into user-created boards called "subreddits", which cover a variety of topics. The dataset is obtained from a data collection forum containing 1.7 billion messages (221 million conversations) from December 2005 to March 2018.[c]

[b] https://www.reddit.com/.
[c] https://files.pushshift.io/reddit/.

Table 1. The number of discussion threads (Disc) picked from 30 different subreddits (SubR).

SubR	Disc	SubR	Disc	SubR	Disc
AskReddit	7	Movies	7	LifeProTip	5
Funny	7	Music	5	Mildlyinte	5
Todayilear	7	Aww	7	DIY	5
Pics	5	Gifs	6	Showerthou	7
Worldnews	7	News	8	Sports	8
IAmA	7	Explainlik	8	Space	6
Announceme	7	Askscience	8	Tifu	5
Videos	9	EarthPorn	7	Jokes	4
Gaming	7	Books	7	InternetIs	9
Blog	7	Television	7	Food	6

After prepossessing, we find that there are 42% posts without any comments and 35% posts with less than or equal to five comments. Most of these discussions only focus on one rather than multiple topics and do not have the topic shift phenomenon, so their topics are easy to be modeled accurately, or we can just use the title of each discussion thread as its topic. In order to prove the effectiveness of our proposed model, we thus filter the posts with the number of replies less than 100, and then randomly picked 200 discussions from 30 different "subreddits". Table 1 lists the details.

There is no category information available for this dataset, so three annotators were asked to label each conversation with the topics, and labels agreed by at least two annotators are used as the ground truth, with a total of 810 topics labeled in this manner. We use a web-based text annotation tool called Tagtog[d] to annotate the topics for each discussion, as Fig. 6 shows.

In the annotation process, the number of topics needs to be set first, and topic assignment of each comment needs to labeled, but the topic set is automatically generated and updated as the labeling work goes on. In addition, the annotation tool will find all the same words across the document and label them, so annotators only need to focus on the words that have not been labeled. In Fig. 6, the labeled words are marked different colors by topics. To simplify the labeling and topic modeling process, each comment is assigned only 1 topic, and the discussion thread is labeled four topics on average to avoid too detailed topic assignment.

We evaluate the performance of the following models, using all their original implementations:

- LDA: The classic Latent Dirichlet Allocation (LDA) model is used as the baseline model. For every dataset, the LDA model is used by setting the hyper parameters $\alpha = 0.1$ and $\beta = 0.01$, and the number of topics $= 70$.[e]

[d] https://www.tagtog.net.
[e] Python library: gensim.models.LdaModel.

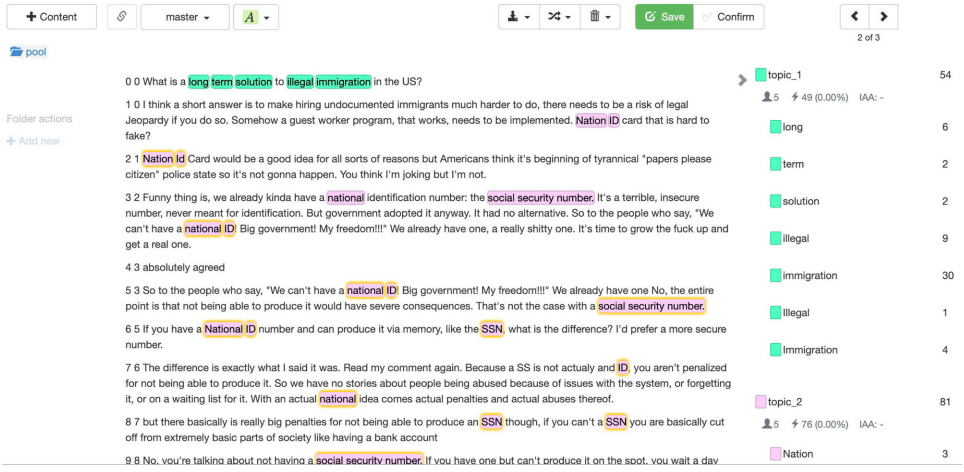

Fig. 6. An example of topic annotation interface of Tagtog.

- PTM: Pseudo document-based Topic Model [16] aggregates short texts against data sparsity. The original implementation with the number of pseudo documents = 1000 and $\lambda = 0.1$.[f]
- BTM: Biterm Topic Model [19] directly models topics of all word pairs (biterms) in each post and explicitly models the word co-occurrence patterns to enhance topic learning. Following the original paper, $\alpha = 50/\mathrm{K}$ and $\beta = 0.01$.[g]
- LeadLDA: Generates words according to topic dependencies derived from conversation trees [30]. A classifier trained to differentiate leader and follower messages is required before using LeadLDA [31], labeled leader and follower messages and CRF are used to obtain the probability distribution of leaders and followers.[h]
- LFTM: Latent Feature LDA [21] incorporates latent feature vector representations of words trained on very large corpora to improve the word-topic mapping learnt on a smaller corpus. Following the paper, the hyper-parameter $\alpha = 0.1$.[i]
- SATM: Self-Aggregation-Based Topic Model [15] aggregates documents and infers topics simultaneously. Following [30], the pseudo-document number is chosen from 100 to 1000 in all evaluations, and the best scores are reported.[j]
- CSATM: We need to select a decreasing sequence to model the weights of the levels used for calculating the popularity score. In this experiment, we use the arithmetic progression with the "sharper" weight distribution because the word distribution of the dataset is pretty sparse and 74% of words show up only once.

[f] http://ipv6.nlsde.buaa.edu.cn/zuoyuan/.
[g] https://github.com/xiaohuiyan/BTM.
[h] https://github.com/girlgunner/leadlda.
[i] https://github.com/datquocnguyen/LFTM.
[j] https://github.com/WHUIR/SATM.

4.2. *Coherence evaluation*

Topic model evaluation is inherently difficult. In previous work, perplexity is a popular metric to evaluate the predictive abilities of topic models using a held-out dataset with unseen words [5]. However, Chang *et al.* [46] have demonstrated that the method does not translate to the actual human interpretability of topics, so the coherence score is widely used to measure the quality of topics [15], assuming that words representing a coherent topic are likely to co-occur within the same document [16]. To reduce the impact of low frequency counts in word co-occurreces, we employ the topic coherence metric called normalized PMI (NPMI) [47]. Given the T most probable words in a topic k, *NPMI* is computed by

$$\mathrm{NPMI}(k) = \frac{2}{T(T-1)} \sum_{1 \le i \le j \le T} \frac{\log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-\log\ p(w_i, w_j))},$$

where $p(w_i)$ and $p(w_i, w_j)$ are the probabilities that word wi occurs, and that the word pair (w_i, w_j) co-occurred estimated by the reference corpus, respectively. T is set to 10 in our experiments. We also use five other confirmation measures to further enhance the comparisons across models.

C_{UCI} is a coherence that is based on a sliding window and the pointwise mutual information (PMI) of all word pairs of the given top words [48]. The word co-occurrence counts are derived using a sliding window with the size 10. For every word pair the PMI is calculated. The arithmetic mean of the PMI values is the result of this coherence

$$C_{\mathrm{UCI}} = \frac{2}{T(T-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}.$$

C_{UMass} is based on document co-occurrence counts, a one-preceding segmentation and a logarithmic conditional probability as confirmation measure [49]. The main idea of this coherence is that the occurrence of every top word should be supported by every top preceding top word. Thus, the probability of a top word to occur should be higher if a document already contains a higher-order top word of the same topic. Therefore, for every word the logarithm of its conditional probability is calculated using every other top word that has a higher order in the ranking of top words as condition. The probabilities are derived using document co-occurrence counts. The single conditional probabilities are summarized using the arithmetic mean

$$C_{\mathrm{UMass}} = \frac{2}{T(T-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \log \frac{p(w_i, w_j)}{p(w_j)}.$$

C_V is based on a sliding window, a one-set segmentation of the top words and an indirect confirmation measure that uses NPMI and the cosinus similarity [50]. This coherence measure retrieves co-occurrence counts for the given words using a sliding window and the window size 110. The counts are used to calculated the NPMI of

Table 2. Averaged coherence, measured by 6 different methods. The top two results are in boldface and italic, respectively.

Measure	Cv	Cp	Cuci	Cumass	NPMI	Ca
LDA	0.370	−0.014	−1.455	−4.186	−0.037	0.137
PTM	0.367	*0.077*	−0.958	−2.783	−0.023	0.091
BTM	0.372	0.015	−1.123	−3.008	−0.022	0.151
leadLDA	**0.396**	0.054	−1.095	−2.962	*0.018*	*0.153*
LFTM	0.359	0.044	−2.012	−3.038	0.008	0.089
SATM	0.368	0.032	−1.086	−3.164	0.007	0.111
CSATM	*0.390*	**0.079**	**−0.915**	*−2.826*	**0.021**	**0.166**

every top word to every other top word, thus, resulting in a set of vectors — one for every top word. The one-set segmentation of the top words leads to the calculation of the similarity between every top word vector and the sum of all top word vectors. As similarity measure the cosinus is used. The coherence is the arithmetic mean of these similarities

$$C_V = \frac{2}{T(T-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathrm{Sim}_{\cos}(w_i, w_j)).$$

C_A is based on a context window, a pairwise comparison of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosinus similarity [50]. This coherence measure retrieves co-occurrence counts for the given words using a context window with the window size 5. The counts are used to calculated the NPMI of every top word to every other top word, thus, resulting in a single vector for every top word. After that the cosinus similarity between all word pairs is calculated. The coherence is the arithmetic mean of these similarities.

C_P is a based on a sliding window, a one-preceding segmentation of the top words and the confirmation measure of Fitelson's coherence [51]. Word co-occurrence counts for the given top words are derived using a sliding window and the window size 70. For every top word, the confirmation to its preceding top word is calculated using the confirmation measure of Fitelson's coherence. The coherence is the arithmetic mean of the confirmation measure results.

Instead of using the collection itself to measure word association — which could reinforce noise or unusual word statistics [52] — we use a large external text data source: an English Wikipedia reference corpus of 8 million documents, and all experiments are conducted on Palmetto platform.[k] The experimental results are given in Table 2.

From the results we observe that the traditional modeling method (LDA) cannot improve the performance of short text topic model. Additionally, we observe that PTM, BTM, LFTM and SATM are almost at the same level. The performance gap among the four is slightly behind LeadLDA and not significant. Recall, LeadLAD

[k] http://aksw.org/Projects/Palmetto.html.

uses labeled messages to help identify potential topical words. CSATM outperforms all baseline models in most cases. We can see that CSATM is competitive against LeadLDA, but doesn't require model training with labeled comments, which saves time and effort.

4.3. *Topic assignment evaluation*

After extracting high-quality topics from the corpus, the assignments of topics to comments should have reasonable accuracy; sometimes it is important to know the "targets" each comment discusses in some downstream applications like stance detection, opinion mining, and so on. In our experiment, we labeled the topic assignments to the top 100 comments in each discussion thread, and compared the performance on CSATM to other models in terms of the accuracy of topic assignment, and the results are given in Fig. 7.

We observe that CSATM achieves much higher accuracy than other models. That's because conventional models cannot deal with noisy comments like emojis, pictures, cursing, and so on in online discussions. CSATM has the ability to find the correct topic distributions of comments through their ancestors in the discussion thread using the proposed topic transitivity property. Take the discussion thread in 1 as an example, there are two topics in that discussion: "concept completely blows your mind" and "all roads work by being connected up". Topics to all comments may be correctly assigned except comment node 9 that is an emoji. Traditional models may fail to assign the right topic for this comment and randomly pick up one. Our model can make the topic of comment 9 correctly assigned by inferring its background information through the conversational structure.

The accuracy of CSATM is still below 0.6 because some of the topics discovered are not correct, so the assignments of topics to comments make no sense in this case. The assignment error of comments leading discussions will affect the correctness of topic assignments of their dependents.

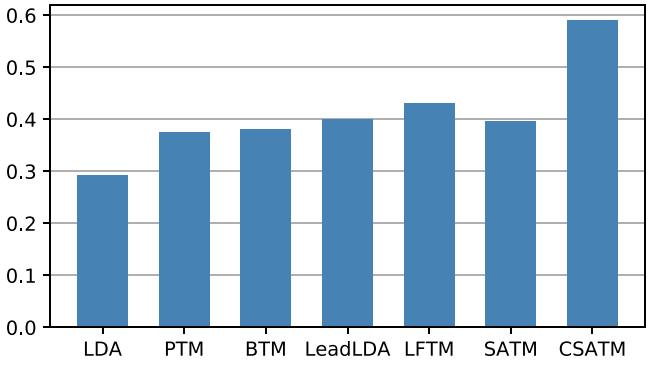

Fig. 7. Accuracy of topic assignments to comments.

5. Case Study

In this section, we use a real case as demo to show the effectiveness of our model. The left box in Fig. 8 is a snippet of an online discussion on the news "Texas serial bomber

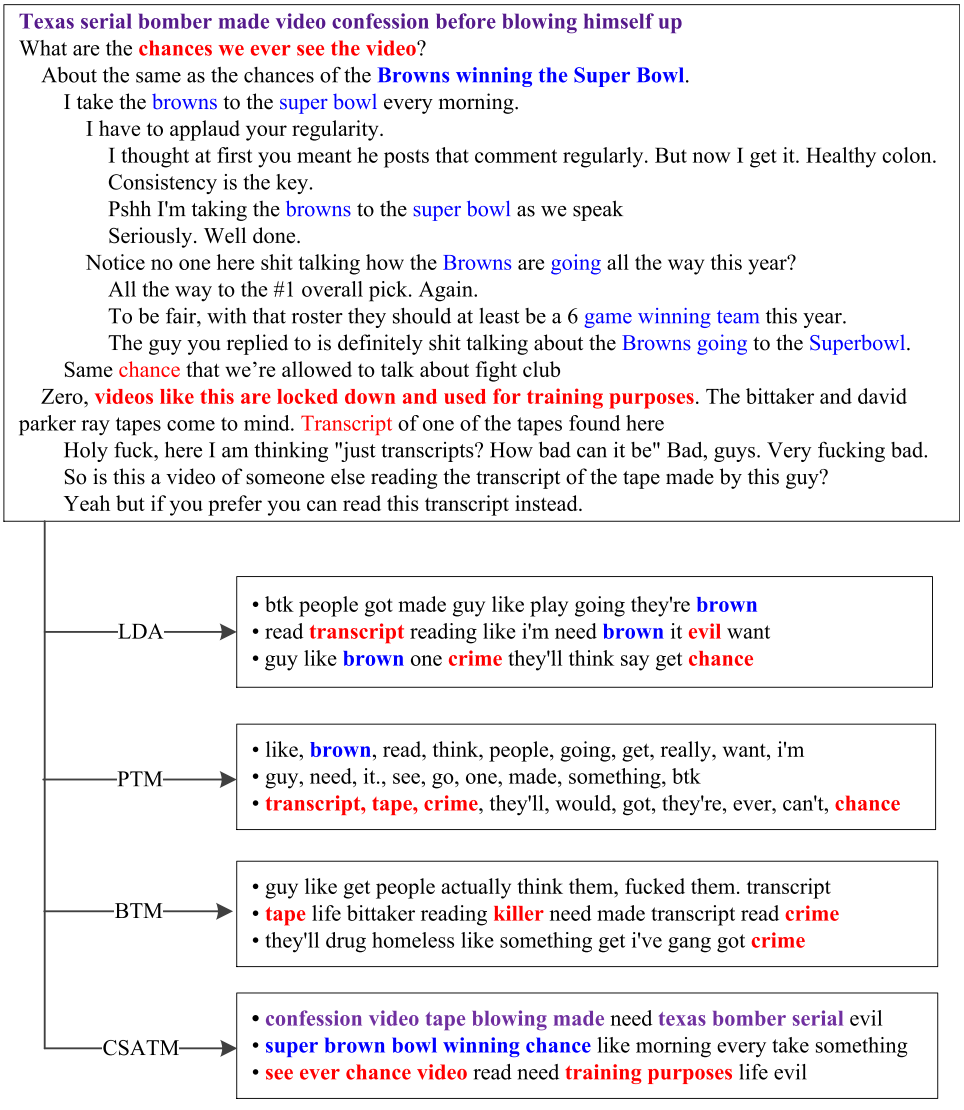

Fig. 8. (Color online) An example thread of user comments on the news: "Texas serial bomber made video confession before blowing himself up"[1] Three topics are bolded and marked by different colors.

[1]https://www.reddit.com/r/news/comments/867njq/texas_serial_bomber_made_video_confession_before/?st=jw0idbj9&sh=fe12e994.

made video confession before blowing himself up". Topics are bolded and marked by different colors. We can see there are basically three topics discussed in this thread: (1) the news title, (2) chance to see the video, (3) Browns win the Super Bowl. This is a very typical and special case, because the topical words are very sparse, and one topic (browns win super bowl) shifts from the main discussion thread.

We set the number as three and use four different topic models to extract the topics: LDA, PTM, BTM and CSATM. We can see that LDA extracted topic 2 and 3, but they are mixed together. PTM extracted topics 2 and 3, but did not capture enough topical words for topic 3. BTM only extracted topic 2. All the three models failed to extract topic 1. Compared to the above three models, CSATM shows great performance by successfully extracted all the three topics with enough topical words. For topics that lead the discussions but their topical words are not repeatedly occurred in the comments and replies, conventional topic models based on word occurrence may not extract such kind of topics successfully, but our proposed model CSATM could deal with this issue. Of course, when the data is not sparse and topic word occurrence is high enough for modeling, CSATM can also achieve good performance by setting the difference of the weight sequence in Eq. (1) to a smaller to value until 1.

6. Conclusion

In this paper, we have proposed the topic "popularity" and "transitivity" intuitions and presented a novel topic model CSATM for online discussions. Conventional works considering only plain text streams is not sufficient enough to summarize noisy discussion trees. CSATM captures the conversational structure as context for topic modeling and topic assignment to each comment, leading to better performance in terms of topic coherence and assignment accuracy. By comparing our proposed model with a number of state-of-the-art baseline models on real word datasets, we have demonstrated competitive results, and the effectiveness of using conversational discourse structure to help in identifying topical content embedded in short and colloquial online discussions. Weight sequence selection may be a little confusing, but it is because of the inherent subjectivity of topic modeling and no uniform metric for the topic extraction.

7. Future Research

The content of online discussions evolve over time, since users keep adding comments or replies to the discussion thread, especially for those discussions with "hot" topics. It is of interest to explicitly model the dynamics of the underlying topics for short text segment collections. Such kind of models can not only capture the newly emerging topics but also keep track of the topic trends of discussion threads. For example, Derek and James are interested in extracting latent thematic patterns in political speeches by developing a dynamic topic model to investigate how the

plenary agenda of the European Parliament has changed over the past terms [53]. Daniel *et al.* seek to uncover the broad trends and facts from social sentences in social networking sites [54]. It might be an interesting research direction to develop dynamic topic models for online discussions with conversational structures [55–57].

There are plentiful downstream applications of our proposed model. For example, it can assistant users to browse a long discussion thread quickly by summarizing the possible topics [58]. Oftentimes, a popular news paper or interesting post can easily accumulate thousands of comments within a short period of time, which makes it difficult for interested users to access and digest information in such data [59, 60]. Therefore, modeling the user-generated comments with respect to different topics and automatically gaining the insight of readers' opinions and attention on the news event will save users' a lot of time.

Acknowledgments

This work was supported by the Ohio Department of Higher Education, the Ohio Federal Research Network and the Wright State Applied Research Corporation under award WSARC-16-00530 (C4ISR: Human-Centered Big Data).

References

[1] J. Hatori, A. Murakami and J. Tsujii, Multi-topical discussion summarization using structured lexical chains and cue words, in *Int. Conf. Intelligent Text Processing and Computational Linguistics*, 2011, pp. 313–327.

[2] R. Dong, Y. Sun, L. Wang, Y. Gu and Y. Zhong, Weakly-guided user stance prediction via joint modeling of content and social interaction, in *Proc. ACM on Conf. Information and Knowledge Management*, 2017, pp. 1249–1258.

[3] A. Guille and C. Favre, Event detection, tracking, and visualization in Twitter: A mention-anomaly-based approach, *Soc. Network Anal. Min.* **5**(1) (2015) 18.

[4] T. Hofmann, Probabilistic latent semantic analysis, in *Proc. Fifteenth Conf. Uncertainty in Artificial Intelligence*, 1999, pp. 289–296.

[5] D. M. Blei, A. Y. Ng and M. I. Jordan, Latent dirichlet allocation, *J. Mach. Learn. Res.* **3** (2003) 993–1022.

[6] L. Hong and B. D. Davison, Empirical study of topic modeling in twitter, in *Proc. First Workshop on Social Media Analytics*, 2010, pp. 80–88.

[7] C. Napoles, A. Pappu and J. Tetreault, Automatically identifying good conversations online (yes, they do exist!), in *Eleventh Int. AAAI Conf. Web and Social Media*, 2017, pp. 628–631.

[8] C. Chen and J. Ren, Forum latent dirichlet allocation for user interest discovery, *Knowl. Based Syst.* **126** (2017) 1–7.

[9] T. Shi, K. Kang, J. Choo and C. K. Reddy, Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations, in *Proc. World Wide Web Conf.*, 2018, pp. 1105–1114.

[10] J. Weng, E.-P. Lim, J. Jiang and Q. He, Twitterrank: Finding topic-sensitive influential twitterers, in *Proc. Third ACM Int. Conf. Web Search and Data Mining*, 2010, pp. 261–270.

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

[12] D. Alvarez-Melis and M. Saveski, Topic modeling in twitter: Aggregating tweets by conversations, in *Tenth Int. AAAI Conf. Web and Social Media*, 2016, pp. 519–522.

[13] D. Ramage, S. Dumais and D. Liebling, Characterizing microblogs with topic models, in *Fourth Int. AAAI Conf. Weblogs and Social Media*, 2010, pp. 130–137.

[14] R. Mehrotra, S. Sanner, W. Buntine and L. Xie, Improving lda topic models for microblogs via tweet pooling and automatic labeling, in *Proc. 36th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2013, pp. 889–892.

[15] X. Quan, C. Kit, Y. Ge and S. J. Pan, Short and sparse text topic modeling via self-aggregation, in *Twenty-Fourth Int. Joint Conf. Artificial Intelligence*, 2015, pp. 2270–2276.

[16] Y. Zuo, J. Wu, H. Zhang, H. Lin, F. Wang, K. Xu and H. Xiong, Topic modeling of short texts: A pseudo-document view, in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 2105–2114.

[17] G. Xun, Y. Li, W. X. Zhao, J. Gao and A. Zhang, A correlated topic model using word embeddings, in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 4207–4213.

[18] B. Shi, W. Lam, S. Jameel, S. Schockaert and K. P. Lai, Jointly learning word embeddings and latent topics, in *Proc. 40th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2017, pp. 375–384.

[19] X. Yan, J. Guo, Y. Lan and X. Cheng, A biterm topic model for short texts, in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1445–1456.

[20] H.-Y. Lu, L.-Y. Xie, N. Kang, C.-J. Wang and J.-Y. Xie, Don't forget the quantifiable relationship between words: Using recurrent neural network for short text topic discovery, in *Thirty-First AAAI Conf. Artificial Intelligence*, 2017, pp. 1192–1198.

[21] D. Q. Nguyen, R. Billingsley, L. Du and M. Johnson, Improving topic models with latent feature word representations, *Trans. Assoc. Comput. Linguis* **3** (2015) 299–313.

[22] X. Li, A. Zhang, C. Li, L. Guo, W. Wang and J. Ouyang, Relational biterm topic model: Short-text topic modeling using word embeddings, *Comput. J.* **62**(3) (2018) 359–372.

[23] Y. Sun and K. Loparo, Knowledge-guided text structuring in clinical trials, in *19th Industrial Conf. Data Mining*, 2019, pp. 211–219.

[24] Y. Sun, X. Liang and K. Loparo, A common gene expression signature analysis method for multiple types of cancer, in *19th Industrial Conf. Data Mining*, 2019, pp. 185–196.

[25] Y. Sun and K. Loparo, Opinion spam detection based on heterogeneous information network, in *IEEE 31st Int. Conf. Tools with Artificial Intelligence*, 2019, pp. 1156–1163.

[26] Y. Sun and K. Loparo, Context aware image annotation in active learning with batch mode, in *IEEE 43rd Annual Computer Software and Applications Conf.*, Vol. 1, 2019, pp. 952–953.

[27] Y. Sun and K. Loparo, A clicked-url feature for transactional query identification, in *2019 IEEE 43rd Annual Computer Software and Applications Conf.*, Vol. 1, 2019, pp. 950–951.

[28] Y. Sun and K. Loparo, Information extraction from free text in clinical trials with knowledge-based distant supervision, in *IEEE 43rd Annual Computer Software and Applications Conf.*, Vol. 1, 2019, pp. 954–955.

[29] Y. Sun, F. Guo, F. Kaffashi, F. J. Jacono, M. DeGeorgia and K. A. Loparo, Insma: An integrated system for multimodal data acquisition and analysis in the intensive care unit, *J. Biomed. Inform.* **106** (2020) 103434.

[30] J. Li, M. Liao, W. Gao, Y. He and K.-F. Wong, Topic extraction from microblog posts using conversation structures, in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 1: Long Papers, 2016.

[31] J. Li, W. Gao, Z. Wei, B. Peng and K.-F. Wong, Using content-level structures for summarizing microblog repost trees, in *Proc. Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2168–2178.

[32] J. Li, Y. Song, Z. Wei and K.-F. Wong, A joint model of conversational discourse and latent topics on microblogs, *Comput. Linguist.* **44**(4) (2018) 719–754.

[33] Y. Sun, R. Kolacinski and K. Loparo, Eliminating search intent bias in learning to rank, in *IEEE 14th Int. Conf. Semantic Computing*, 2020, pp. 108–115.

[34] Y. W. Teh, M. I. Jordan, M. J. Beal and D. M. Blei, Hierarchical dirichlet processes, *J. Am. Stat. Assoc.* **101**(476) (2006) 1566–1581.

[35] J. Paisley, C. Wang, D. M. Blei and M. I. Jordan, Nested hierarchical dirichlet processes, *IEEE Trans. Pattern Anal. Mach. Intell.* **37**(2) (2014) 256–270.

[36] Q. Li, Y. Sun and B. Xue, Complex query recognition based on dynamic learning mechanism, *J. Comput. Inform. Syst.* **8**(20) (2012) 8333–8340.

[37] Q. Li, Y. Zou and Y. Sun, User personalization mechanism in agent-based meta search engine, *J. Comput. Inform. Syst.* **8**(20) (2012) 1–8.

[38] Q. Li, Y. Zou, Y. Sun, J. Xu and B. Xi, An agent-based metasearch engine personalization method, 2016, CN Patent 103,593,413 B.

[39] Q. Li and Y. Sun, An agent based intelligent meta search engine, in *Int. Conf. Web Information Systems and Mining*, 2012, pp. 572–579.

[40] Y. Sun and Q. Li, The research situation and prospect analysis of meta-search engines, in *2nd Int. Conf. Uncertainty Reasoning and Knowledge Engineering*, 2012, pp. 224–229.

[41] Y. Guo, T. Ji, Q. Wang, L. Yu and P. Li, Quantized adversarial training: An iterative quantized local search approach, in *IEEE Int. Conf. Data Mining*, 2019, pp. 1066–1071.

[42] Topic definition, https://www.merriam-webster.com/dictionary/topic, Merriam Webster.

[43] K. Topal, M. Koyuturk and G. Ozsoyoglu, Emotion-and area-driven topic shift analysis in social media discussions, in *IEEE/ACM Int. Conf. Advances in Social Networks Analysis and Mining*, 2016, pp. 510–518.

[44] Y. Sun and K. Loparo, Topic shift detection in online discussions using structural context, in *IEEE 43rd Annual Computer Software and Applications Conf.*, Vol. 1, 2019, pp. 948–949.

[45] A. Park, A. L. Hartzler, J. Huh, G. Hsieh, D. W. McDonald and W. Pratt, "how did we get here?": Topic drift in online health discussions, *J. Med. Internet Res.* **18**(11) (2016) e284.

[46] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber and D. M. Blei, Reading tea leaves: How humans interpret topic models, in *Advances in Neural Information Processing Systems*, 2009, pp. 288–296.

[47] G. Bouma, Normalized (pointwise) mutual information in collocation extraction, *Proc. GSCL*, 2009, pp. 31–40.

[48] D. Newman, J. Lau, K. Grieser and T. Baldwin, Automatic evaluation of topic coherence, in *Human Language Technologies: Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 100–108.

[49] D. Mimno, H. M. Wallach, E. Talley, M. Leenders and A. McCallum, Optimizing semantic coherence in topic models, in *Proc. Conf. Empirical Methods in Natural Language Processing*, 2011, pp. 262–272.

[50] M. Röder, A. Both and A. Hinneburg, Exploring the space of topic coherence measures, in *Proc. Eighth ACM Int. Conf. Web Search and Data Mining*, 2015, pp. 399–408.

[51] B. Fitelson, A probabilistic theory of coherence, *Analysis* **63**(3) (2003) 194–199.

[52] D. Newman, Y. Noh, E. Talley, S. Karimi and T. Baldwin, Evaluating topic models for digital libraries, in *Proc. 10th Annual Joint Conf. Digital Libraries*, 2010, pp. 215–224.

[53] D. Greene and J. P. Cross, Exploring the political agenda of the european parliament using a dynamic topic modeling approach, *Political Anal.* **25**(1) (2017) 77–94.

[54] D. Ramage, E. Rosen, J. Chuang, C. D. Manning and D. A. McFarland, Topic modeling for the social sciences, in *NIPS Workshop on Applications for Topic Models: Text and Beyond*, Vol. 5, 2009, p. 27.

[55] Y. Sun, K. Loparo and R. Kolacinski, Conversational structure aware and context sensitive topic model for online discussions, in *IEEE 14th Int. Conf. Semantic Computing*, 2020, pp. 85–92.

[56] Y. Sun, Topic modeling and spam detection for short text segments in web forums, PhD thesis, OhioLINK ETD Center, 2020.

[57] Y. Sun and K. Loparo, Context aware image annotation in active learning, in *19th Industrial Conf. Data Mining*, Vol. 1, 2019, pp. 251–262.

[58] Q. Li, Y. Sun, Y. Zou, J. Xu and B. Xi, An agent-based intelligent metasearch engine system, 2015, CN Patent 102,902,800 B.

[59] Q. Li, Y. Zou and Y. Sun, Ontology based user personalization mechanism in meta search engine, in *2nd Int. Conf. Uncertainty Reasoning and Knowledge Engineering*, 2012, pp. 230–234.

[60] Y. Sun, X. Cai and K. Loparo, Learning-based adaptation framework for elastic software systems, in *IEEE 31st Int. Conf. Software Engineering & Knowledge Engineering*, Vol. 1, 2019, pp. 281–286.