SN Computer Science (2021) 2:256
https://doi.org/10.1007/s42979-021-00649-x

ORIGINAL RESEARCH

LJST: A Semi-supervised Joint Sentiment-Topic Model for Short Texts

Ayan Sengupta[1] · **Suman Roy**[2] · **Gaurav Ranjan**[2]

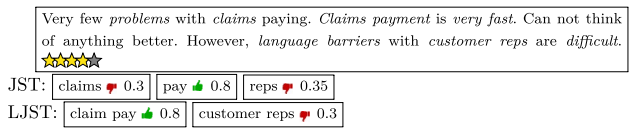
Received: 21 April 2020 / Accepted: 16 April 2021 / Published online: 6 May 2021

Abstract
Several methods on simultaneous detection of sentiment and topics have been proposed to obtain subjective information such as opinion, attitude and feelings expressed in texts. Most of the techniques fail to produce desired results for short texts. In this paper, we propose LJST, a labeled joint sentiment-topic model particularly for short texts. It uses a probabilistic framework based on latent Dirichlet allocation. LJST is semi-supervised—it predicts the sentiment values for unlabeled texts in presence of a partially labeled texts with sentiment values. To address the sparsity problem in short text, we modify LJST and introduce Bi-LJST, which uses bi-terms (all possible pairs of words in a document) in place of unigrams for learning the topics by directly generating word co-occurrence patterns in each text and expressing the topics in terms of these patterns. Specifically, we have proposed a semi-supervised approach of extracting joint sentiment-topic model for short texts by incorporating bi-terms. Extensive experiments on three real-world datasets show that our methods perform consistently better than three other baselines in terms of document-level and topic-level sentiment prediction, and topic discovery—LJST using bi-term models outperforms the best baseline by producing 12% lower RMSE for document-level sentiment prediction and 6% higher F1 score for topic-sentiment prediction.

Keywords Topic models · Sentiment extraction · Joint sentiment topic models · Short texts · Bi-terms

Introduction

Short text is a popular medium of communication in the online social networks that are spread across the Internet and appear abundant in different applications. Mining short texts is essential to identify the sentiment expressed by the users about certain entities (products, services, political figures, etc.). The objective of sentiment analysis is to classify the sentiments into positive, negative or neutral classes. However, in many practical scenarios, it may be required to discover both topic and sentiment jointly. For example, in target dependent (or topic-specific) sentiment analysis [14], simultaneous detection of targets/topics is required along with the sentiment label. There have been few attempts to predict both sentiment and topics simultaneously [19, 22, 26, 31]; among which joint sentiment-topic (JST) extraction approach is quite popular. To see how JST works, let us consider the following example of a review:



JST will discover the following topics: `claims`, `pay`, `reps`, etc. Using an appropriate sentiment lexicon, JST will also detect sentiment values of the topics as shown above without considering any external sentiment labels (star rating). The major limitations of JST-based approaches are two-fold. Firstly, most of these techniques deal with only unlabeled data; thus they are unable to incorporate external labels such as the ratings given by the users, ground-truth labels obtained from the annotators, etc. We hypothesize

✉ Ayan Sengupta
ayan_sengupta@optum.com

Suman Roy
suman.roy@optum.com

Gaurav Ranjan
gauravranjan@optum.com

[1] Optum Global Solutions (India) Pvt Ltd (UnitedHealth Group), Oxygen Business Park, Sector 144, Noida, Uttar Pradesh 201304, India

[2] Optum Global Solutions (India) Pvt Ltd (UnitedHealth Group), Sarjapur-Marathahalli Outer Ring Road, Bangalore 560 103, India

that external labels often play an important role in determining the sentiment and topics jointly. For instance, in the above example, the 4-star rating given by the customer can be incorporated to better identify the sentiment (see the results obtained from our LJST model for the same example mentioned above). Also, by using context based information through skip-gram modeling, one can extract more meaningful topics as witnessed in this case. Secondly, in the absence of appropriate lexicon, JST will fail to extract topics and associated sentiments.

In this paper, we present labeled joint sentiment topic model (LJST), a novel semi-supervised framework that jointly discovers topics and sentiments for short texts by overcoming both the problems mentioned above in presence of both labeled (with discrete/continuous values) and unlabeled texts. LJST is motivated by the work of weakly supervised joint topic-sentiment model [19], an extension of the classic topic model based on latent Dirichlet allocation (LDA) [6]. In LJST we construct an additional sentiment layer on top of LDA by assuming that sentiments are generated based on topic distributions [19], and words are generated by conditioning on the topic-sentiment pairs. We also discover topics and predict sentiment for short texts by drawing bi-terms (we define bi-terms as a pair of words appearing in any order in the same sentence), instead of unigrams according to a topic-biterm distribution in presence of a collection of labeled short texts. This way we address the sparsity problem in short texts by modifying the original framework to learn bi-terms (that can capture word co-occurrence pattern) from sentiment topic distribution. We call it bi-term labeled sentiment topic model (Bi-LJST). Its performance turns out to be better than that of LJST in terms of the quality of both sentiment and topics extracted.

We conduct a set of experiments on two publicly available datasets and one internal dataset. Our methods are compared with three state-of-the-art baselines. We show the superiority of our methods on both sentiment prediction and topic discovery. In the former case, Bi-LJST beats the best baseline by more than 12% higher RMSE for document-level sentiment prediction, and by 6% higher F1 score for topic-specific sentiment prediction. In the latter setting, the improvement is 24% higher than the best baseline. We also vary different parameters of the models to show how our models respond to the variation.

Motivation and real-world application: This work stems from a very practical real-life use case scenario. We have had a lot of feedback texts (which are short) arising in healthcare services. While some of these feedback texts are labeled with sentiment values, the rest are not assigned any label. The requirement from the business was to label the latter texts with sentiment values as well as to discover topics from the collection, topic distribution for each text, and finally assign sentiment values to each topic appearing in a text. Towards that, we propose a novel solution which can identify topics from documents and associate sentiments with them. Further, our invention can be used for capturing overall sentiment score for the unlabeled texts. The core idea relies on the fact that users before assigning a rating to a feedback text, decompose the feedback in few topics and assign the final rating based on those topics. Other existing approaches do not consider the users' overall ratings to identify the topics and sentiment rating of topics which leads to poor results. Additionally, most of the existing solutions use only 0/1 sentiment rating, whereas in many real-life applications such as customer feedback, product rating etc., users provide discrete valued ratings within a range (like between 0 and 5 or 0 and 10). Several existing customer relationship management (CRM) tools seek ratings (these may be discrete) on various aspects of any service/offering and calculate the average of this ratings as the final for the overall feedback, which may be real-valued. Thus, it is of utmost importance to build a system that can handle both the types of sentiment values, discrete and real. In our model we discretize real-valued sentiments into fine-grained discrete classes. By doing so, LJST allows us to tackle any discrete as well as any real valued (continuous) sentiment labels.

Contribution of our work: We propose an approach of extracting sentiment and topic jointly, which provides advantages over other joint sentiment-topic models—it generates a joint sentiment-topic model by using bi-terms which is well-suited for short texts; it works in the presence of both labeled (with sentiment score) and unlabeled texts and produces sentiment values for unlabeled texts, and as well for topics and words appearing in each topic. We empirically show the merits of our approach in comparison to other competitive approaches with respect to the above-mentioned points of view by carrying out an extensive experimentation. This model also is quite relevant to many practical applications. For example, voice of customer (VOC) analysis requires identification of key critical issues and mechanism to prioritize critical issues for operational purposes. Such critical issues can be discovered by examining extracted topics along with their sentiment scores (both local and global). Customer feedback analysis is also necessary for the improvement of NPS (Net Promoter Score) which can be facilitated through the computation of sentiment score of unlabeled documents.

Organization of the paper: The paper is organized as follows. In "Related Work", we summarize the literature relevant to our work. We describe our proposed joint topic sentiment topic model in "Learning Sentiment and Topic Jointly". We narrate the strategy of setting different hyper-parameters in our experimentation in "Hyperparameter Setting". The data sets considered for experimentation are mentioned in "Datasets and Baselines" along with the narration of baseline methods. We produce the results of our experiments in "Experimental Results". We conclude in last section.

For the purpose of reproducible research, we have made all the codes, and datasets public at Ref. [37].

Related Work

We briefly describe related studies in three parts: sentiment analysis, topic discovery, and joint sentiment-topic extraction. Due to the abundance of literature on sentiment analysis and topic modeling, we restrict ourselves to those studies which we deem to be pertinent to our work. Readers are referred to the following survey papers for other prior art [1, 10, 17, 20, 43].

Sentiment analysis: Document-level sentiment analysis has been a widely studied problem [29]. Turney [39] propose an unsupervised learning model for determining opinion polarity of reviews. Pang et al. [30] use classical techniques for semantically classifying movie reviews; however, it fails to perform well in the problem of sentiment classification. To overcome this, the authors further propose a semi-supervised model which uses a two-step model comprising of subjectivity detection and minimum cut finding in graphs [28]. Whitelaw et al. [40] use SVM to train on the combination of different types of appraisal groups and features for sentiment analysis. There are other notable studies on sentiment detection, e.g., Ref. [18] which works on combining individual word-level sentiments, Ref. [44] which uses conditional random field (CRF), Ref. [3] which combines lexicon-based and corpus-based approaches and so on.

Topic discovery: Majority of research work such as PLSA [15], LDA [6], supervised LDA (sLDA) [5], NMF [41] etc., focus on discovering topics from lengthy texts. However, when it comes to discovering topics from short texts such as tweets, customer reviews [9, 16], they perform poorly. Better quality topic models for short texts can be extracted using bi-terms [42]. There have been a few attempts to extract topics from short texts primarily using NMF-based techniques. Examples include [8] which uses correlation matrix built for each pair of unique terms, [27] which uses tweet text similarity and interaction measure, [33] which uses heuristics on top of the former approach, [35] which uses context based information, Cheng et al. which have compared basic LDA and NMF by extracting topics with different experimental settings on several public short text datasets [7] to name a few.

Joint sentiment-topic extraction: All the approaches mentioned above deal with the problems of sentiment extraction and topic discovery independently. Topic-sentiment model (TSM) [21] is the first attempt to solve this as a combined problem. As TSM is primarily based on probabilistic latent semantic indexing (pLSI) [15], it suffers from two common drawbacks: problem with inferring topics for new document and over-fitting. To overcome these, Lin et al. [19] propose a weakly supervised hierarchical Bayesian model (JST). As JST model ensures topic generation to be conditioned on sentiment labels, the authors introduce another model, called reverse-JST (RJST) in which sentiment generation depends on the topic. Multi-grain LDA (MG-LDA) [38], an extension of LDA, builds topics that represent gradable aspects of objects from the online reviews of users. Nguyen and Shirai [26] propose topic sentiment latent Dirichlet allocation (TSLDA), a new topic model that could capture the topic and sentiment simultaneously. In presence of small training corpus and/or short documents, Fu et al. [13] present a novel topic sentiment joint model, called weakly supervised topic sentiment joint model with word embeddings (WS-TSWE), which takes into account both word embeddings and HowNet lexicon simultaneously, and improves the topic identification and sentiment recognition. Ali et al. [2] propose an ontology and latent Dirichlet allocation (OLDA) based topic modeling and word embedding approach for sentiment classification, which retrieves transportation content from social networks, removes irrelevant content to extract meaningful information, and generates topics and features from extracted data. However, all these models do not discover fine-grained dependency between topics and sentiments. To address this, hidden topic-sentiment model (HTSM) is introduced that explicitly captures topic coherence and sentiment consistency from opinionated texts [31]. Along these lines, Fu et al. [12] propose a novel probabilistic model framework based on the non-parametric hierarchical Dirichlet process (HDP) topic model, called non-parametric joint sentiment topic mixture model (NJST), which builds a sentiment level on top of HDP topic model and discovers sentiment and topics simultaneously from reviews. This can tackle the challenging problem of capturing the mixture of sentiment and topics simultaneously in reviews.

Comparison of LJST model with existing models: We discuss how the proposed LJST model differs from the previously formulated joint sentiment-topic models. Similar to RJST [19], our model (LJST) makes sentiment label generation being conditioned on topic generation. However, unlike others, LJST does not require any prior knowledge in the form of word lexicon. Moreover, it can easily incorporate both labeled and unlabeled documents. Also, none of the competing methods are suitable for short texts in contrast to our model, which can be applied to short texts through the introduction of bi-terms. It should be noted that most of the existing models accept only unlabeled texts as input. Table 1 shows a comparison of LJST with existing models w.r.t. different dimensions of the model. Furthermore, it provides certain advantages over the existing methods of joint sentiment topic extraction as can be seen from the comparison between different relevant methods in Table 1—LJST produces per-topic sentiment score (global assignment), per topic per-word sentiment label (for word appearing in a topic, which is called a local assignment) and overall sentiment score for a document.

Table 1 Comparison of LJST and other baseline methods w.r.t different dimensions

Model	Input data	Lexicon needed?	Output			
			T–S	T dist. over W	T polarity	Doc. polarity
JST	UL	Yes	T under S	Global	Doc-level	Yes
TSM	UL	Yes	T–S pair	Local	Global	No
RJST	UL	Yes	T–S pair	Local	Doc-level	No
LJST	UL, L	No	T–S pair	Local	Doc-level	Yes

T topic, *S* sentiment, *W* word, *Doc* document, *UL* unlabeled, *L* labeled

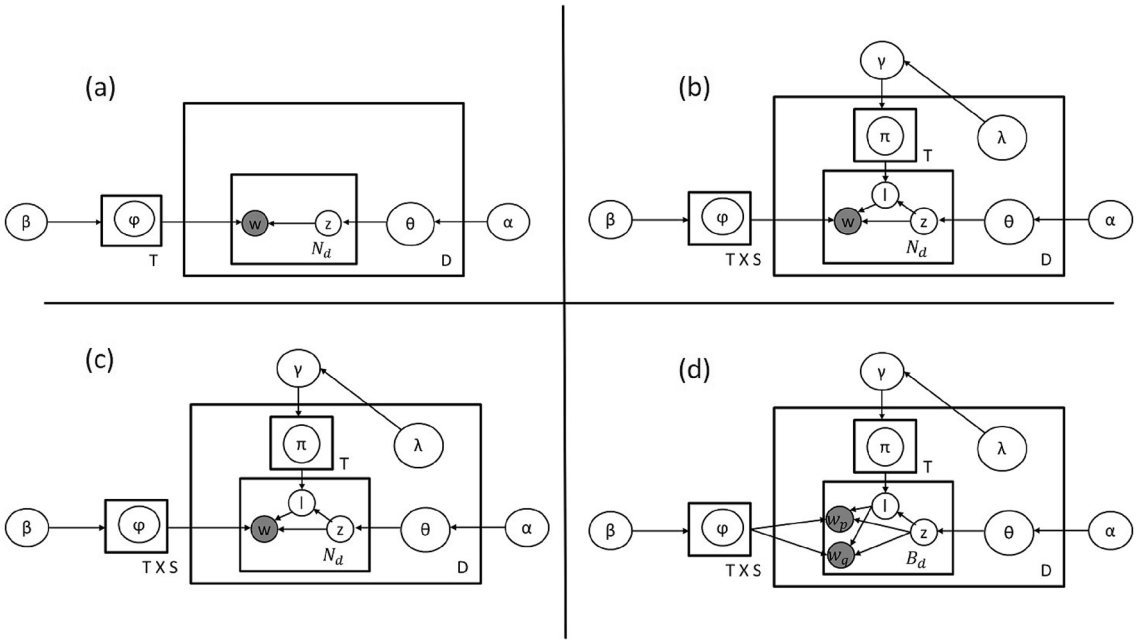


Fig. 1 **a** LDA model; **b** RJST model; **c** LJST model with unigram; **d** LJST model with bi-term (Bi-LJST)

Learning Sentiment and Topic Jointly

Our model is inspired by LDA which represents a document as a mixture of topics and a topic as a probability distribution of words [4, 6] (see Fig. 1a). There are three hierarchical layers in LDA—documents, topics and words, in which topics are associated with documents and so words with topics. To model document sentiment and topics jointly, existing JST models have used an additional sentiment layer between document and topic [11, 19]— new sentiment labels are associated with the document, below which topics are associated with the sentiment labels, and words are associated with both sentiment labels and topics.

LJST: Our Proposed Model

A glossary of notations used throughout the paper is listed in Table 2. Let $\mathcal{C} = \{d_1, d_2, \ldots, d_D\}$ denote a collection of D documents. A document $d = w_1, w_2, \ldots, w_{N_d}$ is represented by a sequence of N_d words. Distinct words are indexed in a vocabulary $\mathcal{V}$ of size V. Also let S and T be the number of distinct sentiment labels and topics respectively. As discussed in the subsequent sections, to sample from the posterior distribution, we need to ensure discrete sentiment values. Hence, we scale our real-valued sentiment scores into [0, 1] and partition this range of real numbers into $S-1$ equal intervals having S discrete points. This allows to define a document-specific label projection vector $\mathbf{L}^{(d)}$ of dimension S as:

$$L_i^{(d)} = \begin{cases} 1 & \text{if } \lambda^d = i \\ 0 & \text{otherwise} \end{cases}.$$

In other words, the ith entry of $\mathbf{L}^d$ is 1 if the label of document d is i. We approximate it by $L^{(d)} \leftarrow L^{(d)} + \epsilon$, $0 < \epsilon < 1$.[1] Generating a word w_i in document d is a three-stage procedure, as shown in Fig. 1c. First, a topic j is chosen from

[1] The parameter ϵ is used to forcibly assign non-zero values to labels.

Table 2 Notations used in the paper

Variable	Type	Description
D	Integer	Number of documents
T	Integer	Number of topics
S	Integer	Number of sentiment class
V	Integer	Number of words in vocabulary
B	Integer	Number of bi-terms in bi-term vocab
N_d	Integer	Number of words in document d
$\boldsymbol{\alpha}$	Vector	Document-topic prior of size T
α_j	Positive real	Document-topic prior for topic j
β	Positive real	Topic-sentiment-word prior
$\boldsymbol{\gamma}$	Matrix	Document-topic-sentiment prior of size $D \times S$
$\boldsymbol{\gamma_d}$	Vector	Document-topic-sentiment prior for document d
$\gamma_{d,j}$	Positive real	Document-topic-sentiment prior for document d and topic j
θ	Matrix	Document-topic probability distribution of size $D \times T$
θ_d	Vector	Topic proportions of document d
π	Tensor	Document-topic-sentiment distribution of size $D \times T \times S$
$\pi_{d,j}$	Vector	Sentiment proportions of document d and topic j
φ	Tensor	Topic-sentiment-word distribution of size $T \times S \times V$
$\varphi_{j,k}$	Vector	Word proportions of topic j and sentiment k
$N_{d,j,k}$	Non-negative integer	Number of words in document d for which sentiment k is assigned to topic j
$N_{j,k}$	Non-negative integer	Number of words for which sentiment k is assigned to topic j
$N_{d,j}$	Non-negative integer	Number of times a word from document d is assigned to topic j
$N_{j,k,v}$	Non-negative integer	Number of times topic j and sentiment k is assigned to word v

a per-document topic distribution θ_d. Following this, a sentiment label l is chosen from sentiment distribution $\pi_{d,j}$, which is conditioned on the sampled topic j. Finally, a word is drawn from the per-corpus word distribution conditioned on both topics and sentiment labels $\varphi_{j,l}$. The steps for the generative process in LJST as shown in Fig. 1c are formalized below:

1. For each document d
   Generate $\theta_d \sim \text{Dir}(\boldsymbol{\alpha})$;
2. For each document d and topic $j \in \{1, 2, \ldots T\}$
   Choose $\pi_{d,j} \sim \text{Dir}(\boldsymbol{\gamma}^{(d)})$, $\boldsymbol{\gamma}^{(d)} = \gamma \times \mathbf{L}^{(d)}$;
3. For each topic $j \in \{1, 2, \ldots T\}$ and sentiment label $l \in \{1, 2, \ldots S\}$
   Choose $\varphi_{j,l} \sim \text{Dir}(\beta)$;
4. For each word w_i in document d
   - **(a)** Choose topic $z_i \sim \text{Mult}(\theta_d)$;
   - **(b)** Choose sentiment label $l_i \sim \text{Mult}(\pi_{d,z_i})$;
   - **(c)** Choose word $w_i \sim \text{Mult}(\varphi_{z_i,l_i})$, a multinomial distribution over words conditioned on sentiment level l_i and topic z_i.

Here, $\boldsymbol{\alpha}$ and β are hyperparameters—the former can be interpreted as the prior observation count, denoting the number of times topic j is associated with document d, and the latter can be interpreted as the number of times words sampled from topic j are associated with sentiment label l_i, before observing the actual words. Dir() is the Dirichlet distribution. Similarly, the hyperparameter $\boldsymbol{\gamma}$ indicates the prior observation number that counts how many times a document d will have the label l before any word from the document is observed. We also use the vector $\mathbf{L}^{(d)}$ to project the parameter vector of the Dirichlet document sentiment prior $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_S)$ to a lower dimensional vector [32]:

$$\boldsymbol{\gamma}^{(d)} = \boldsymbol{\gamma} \times \mathbf{L}^{(d)} = \begin{cases} (1+\epsilon)2\gamma & \text{if } \lambda^d = i \\ \epsilon\gamma & \text{otherwise} \end{cases}.$$

In our implementation (to be discussed in "Hyperparameter Setting"), we have used asymmetric priors $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ and symmetric prior β. We need to infer three sets of latent variables—per-document topic distribution θ, per-document topic specific sentiment distribution π, and per-corpus joint topic-sentiment word distribution φ. Detailed derivations and proofs are available in "Appendix".

Model Inference

To infer θ, π and φ, we first estimate the posterior distribution over z and l, which are the assignment of word tokens to topics and sentiment labels, respectively for each document d. The sampling distribution word $\mathbf{w}$ given the remaining

topics is $p(z_t = j, l_t = k \mid \mathbf{w}, \mathbf{z^{-t}}, \mathbf{l^{-t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma})$, where $\mathbf{z^{-t}}$ and $\mathbf{l^{-t}}$ are the vectors of assignments of topics and sentiments respectively for all the words in the collection except for the word at the tth position in document d.

The joint probability of the words, topics and sentiment labels can be decomposed as follows:

$$p(\mathbf{w}, \mathbf{z}, \mathbf{l}) = p(\mathbf{w} \mid \mathbf{l}, \mathbf{z}) p(\mathbf{l}, \mathbf{z}) = p(\mathbf{w} \mid \mathbf{l}, \mathbf{z}) p(\mathbf{l}|\mathbf{z}) p(\mathbf{z}). \tag{1}$$

The first term of Eq. (1) is obtained by integrating w.r.t. φ:

$$p(\mathbf{w} \mid \mathbf{l}, \mathbf{z}) = \left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V} \right)^{T \times S} \prod_j \prod_k \frac{\prod_i \Gamma(N_{j,k,i} + \beta)}{\Gamma(N_{j,k} + V\beta)}, \tag{2}$$

where $N_{j,k,i}$ is the number of times word i appears in topic j with sentiment label k, and $N_{j,k}$ is the number of times words are assigned to topic j with sentiment label k.

The second term of Eq. (1) is obtained by integrating w.r.t. π:

$$p(\mathbf{l}|\mathbf{z}) = \left(\frac{\Gamma(\sum_{k=1}^{S} \gamma_{d,k})}{\prod_{k=1}^{S} \Gamma(\gamma_{d,k})} \right)^{D \times T} \prod_d \prod_j \frac{\prod_k \Gamma(N_{d,j,k} + \gamma_{d,k})}{\Gamma(N_{d,j} + \sum_k \gamma_{d,k})}, \tag{3}$$

where, $N_{d,j,k}$ is the number of times a word from document d is associated with topic j and sentiment label k, and $N_{d,j}$ is the number of times topic j is assigned to some word tokens in document d.

Similarly, we write the third term of Eq. (1) by integrating w.r.t. θ:

$$p(\mathbf{z}) = \left(\frac{\Gamma(\sum_{j=1}^{T} \alpha_j)}{\prod_{j=1}^{T} \Gamma(\alpha_j)} \right)^{D} \prod_d \frac{\prod_j \Gamma(N_{d,j} + \alpha_j)}{\Gamma(N_d + \sum_{j=1}^{T} \alpha_j)},$$

where N_d is the total number of words in document d.

We employ Gibbs sampling to estimate the posterior distribution by sampling the variables of interest z_t and l_t here, for word w_t from the distribution over the variables, given the current values of all other variables and data. We now compute the joint probability distribution in Eq. (1):

$$p\left(z_t = j, l_t = k \mid w_t, \mathbf{z^{-t}}, \mathbf{l^{-t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}\right) \propto \frac{N^{-t}_{j,k,w_t} + \beta}{N^{-t}_{j,k} + V\beta} \cdot \frac{N^{-t}_{d,j,k} + \gamma_{d,k}}{N^{-t}_{d,j} + \sum_k \gamma_{d,k}} \cdot \frac{N^{-t}_{d,j} + \alpha_j}{N^{-t}_{d} + \sum_j \alpha_j}. \tag{4}$$

We obtain samples from the Markov chain which are then used to approximate the per-corpus topic-sentiment word distribution:

$$\varphi_{j,k,i} = \frac{N_{j,k,i} + \beta}{N_{j,k} + V\beta}. \tag{5}$$

The approximate per-document topic specific sentiment distribution is,

$$\pi_{d,j,k} = \frac{N_{d,j,k} + \gamma_{d,k}}{N_{d,j} + \sum_k \gamma_{d,k}}. \tag{6}$$

Finally, we approximate per-document topic distribution as,

$$\theta_{d,j} = \frac{N_{d,j} + \alpha_j}{N_d + \sum_j \alpha_j}. \tag{7}$$

Algorithm 1 shows the pseudo-code for the Gibbs sampling procedure of LJST.

Algorithm 1: Gibbs sampling procedure for LJST

Input : $\boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}^{(d)}$, Corpus (labeled and unlabeled)
Initialization: Initialize matrix $\boldsymbol{\Theta}_{D \times T}$, tensor $\boldsymbol{\Pi}_{D \times T \times S}$, tensor $\Phi_{T \times S \times V}$;
for *iter = 1 to* max *Gibbs sampling iterations* **do**
  for *all documents* $d \in \{1, 2, \ldots, D\}$ **do**
    for *all words* w_t, $t \in \{1, 2, \ldots, N_d\}$ **do**
      Exclude w_t associated with topic j and sentiment label k and compute $N_{j,k,i}, N_{j,k}, N_{d,j,k}, N_{d,j}$, and N_d;
      Sample a new topic-sentiment pair $\bar{z}$ and $\bar{k}$ using Eq. 4;
      Update variables $N_{j,k,i}, N_{j,k}, N_{d,j,k}, N_{d,j}$, and N_d using the new topic label $\bar{z}$ and sentiment label $\bar{k}$;
    end
  end
  for *every 5 iterations* **do**
    Update hyperparameter $\boldsymbol{\alpha}$ with the maximum likelihood estimation according to Eq. 8
  end
  if *number of iterations* = max *Gibbs sampling iterations* **then**
    Update Θ, Π and Φ with new sampling results given by Eqs 7, 6 and 5
  else
    True
  end
end

We update Dirichlet parameter $\boldsymbol{\alpha}$ using Minka's fixed point iteration scheme [25]. For each topic j, we update α_j as:

$$\alpha_j \leftarrow \alpha_j \cdot \frac{\sum_d \Psi\left(N_{d,j} + \alpha_j\right) - D \cdot \Psi\left(\alpha_j\right)}{\sum_d \Psi\left(\sum_j N_{d,j} + \sum_j \alpha_j\right) - D \cdot \Psi\left(\sum_j \alpha_j\right)}, \tag{8}$$

where Ψ is the derivative of gamma function: $\Psi(x) = \frac{d}{dx}\Gamma(x)$.

Handling Sparsity in LJST

Like other LDA based models, LJST also suffers from data sparsity for short texts. To address this, we modify LJST by considering bi-terms as observed variables rather than unigrams. We call this model bi-term labeled joint sentiment topic (Bi-LJST) model. To alleviate the sparsity problem in short texts Yan et al. propose bi-term topic model (BTM) to model topic in short texts in Ref. [42]. We consider any two distinct words occurring in a short text as a candidate bi-term. Recall a bi-gram is a pair of consecutive words occurring in a text. Unlike bi-grams, bi-terms allow all word pairs within a document irrespective of whether they are consecutive or not. For example - a text saying "great product loved camera", consists of 6 bi-terms, namely - "great product", "great loved", "great camera", "product loved", "product camera" and "loved camera". We take each document as an individual context or use a skip-gram model by choosing a window of specific size around a word as its context [23]. Using a window of size 2 as context will allow us to drop the bi-term "great camera" from the set, as the distance between these two words in the text is 3. The key idea behind Bi-LJST is to learn topics over short texts based on the aggregated bi-terms in the corpus. Figure 1d shows the framework of Bi-LJST. Specifically, we assume that the whole corpus is a mixture of topics in which each bi-term is drawn from a specific topic independently. The probability that a bi-term is drawn from a topic is captured by the assumption that both words in the bi-term are drawn from the same topic. The generative process of Bi-LJST is described as below:

1. For each document d

Generate $\theta_d \sim \text{Dir}(\boldsymbol{\alpha})$;

2. For each document d and topic $j \in \{1, 2, \ldots T\}$

Choose $\pi_{d,j} \sim \text{Dir}(\boldsymbol{\gamma}^{(d)})$, $\boldsymbol{\gamma}^{(d)} = \gamma \times \mathbf{L}^{(d)}$;

3. For each topic $j \in \{1, 2, \ldots T\}$ and sentiment label $l \in \{1, 2, \ldots S\}$

Choose $\varphi_{j,l} \sim \text{Dir}(\beta)$;

4. For each bi-term $b_t = (w_p, w_q)$ in document d

(a) Choose topic $z_t \sim \text{Mult}(\theta_d)$;

(b) Choose sentiment label $l_t \sim \text{Mult}(\pi_{d,z_t})$;

(c) Choose a pair of words $w_p \sim \text{Mult}(\varphi_{z_t,l_t})$ and $w_q \sim \text{Mult}(\varphi_{z_t,l_t})$ independently.

There are some changes in the probability distribution for Bi-LJST. Let $\mathcal{B}$, B and B_d denote the vocabulary of bi-terms, cardinality of $\mathcal{B}$ and the number of bi-terms in the document d. Further, assume that a bi-term is represented as $b_t = (w_p, w_q)$. The joint probability of bi-terms, topics and sentiment labels can be captured by,

$$p\left(b_t, \mathbf{z}, \mathbf{l}\right) = p\left(w_p \mid \mathbf{l}, \mathbf{z}\right) p\left(w_q \mid \mathbf{l}, \mathbf{z}\right) p(\mathbf{l}|\mathbf{z}) p(\mathbf{z}). \tag{9}$$

The first two terms in Eq. (9) are obtained by integrating w.r.t. φ as done in Eq. (1):

$$p(\mathbf{b} \mid \mathbf{l}, \mathbf{z}) \propto \prod_j \prod_k \frac{\prod_p \Gamma\left(N_{j,k,w_p} + \beta\right)}{\Gamma\left(N_{j,k} + V\beta\right)} \frac{\prod_q \Gamma\left(N_{j,k,w_q} + \beta\right)}{\Gamma\left(N_{j,k} + V\beta\right)}. \tag{10}$$

The posterior distribution is obtained by sampling the variables of interest z and l after marginalizing at φ, θ and π:

$$p\left(z_t = j, l_t = k \mid b_t, \mathbf{z}^{-\mathbf{t}}, \mathbf{l}^{-\mathbf{t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}\right) \propto \frac{\left(N^{-t}_{j,k,w_p} + \beta\right) \cdot \left(N^{-t}_{j,k,w_q} + \beta\right)}{\left(N^{-t}_{j,k} + V\beta\right)^2} \cdot \frac{N^{-t}_{d,j,k} + \gamma_{d,k}}{N^{-t}_{d,j} + \sum_k \gamma_{d,k}} \cdot \frac{N^{-t}_{d,j} + \alpha_j}{N^{-t}_{d} + \sum_j \alpha_j}. \tag{11}$$

Other distributions in Eqs. (5), (6) and (7) can be computed in the same manner as for LJST. Similar to Algorithm 1, we use Gibbs sampling procedure for Bi-LJST for parameter estimation. Algorithm 2 shows the pseudo-code for the Gibbs sampling procedure of Bi-LJST.

Table 3 Statistics of the datasets

Dataset	Corpus size	Unigram count	Bigram count	Mean len.	Sentiment type	Sentiment range
IFD	8086	1138	203,353	14.42	Continuous	[−5, 5]
STF	17,227	4897	216,460	8.69	Continuous	[0, 10]
YRD	4633	2500	424,856	28.38	Discrete	{1, 2, 3, 4, 5}
$\text{YRD}_{\text{short}}$	897	2382	35,630	10.37	Discrete	{1, 2, 3, 4, 5}

Algorithm 2: Gibbs sampling procedure for Bi-LJST

Input : $\boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}^{(d)}$, Corpus (labeled and unlabeled)
Initialization: Initialize matrix $\boldsymbol{\Theta}_{D\times T}$, tensor $\boldsymbol{\Pi}_{D\times T\times S}$, tensor $\Phi_{T\times S\times V}$;
for *iter = 1 to* max *Gibbs sampling iterations* **do**
  for *all documents* $d \in \{1, 2, \ldots, D\}$ **do**
    for *all bi-terms* $(w_p, w_q) = b_t$, $t \in \{1, 2, \ldots, B_d\}$ **do**
      Exclude w_p and w_q associated with topic j and sentiment label k and compute $N_{j,k,i}, N_{j,k}, N_{d,j,k}, N_{d,j}$, and N_d;
      Sample a new topic-sentiment pair $\bar{z}$ and $\bar{k}$ using Eq. 11;
      Update variables $N_{j,k,i}, N_{j,k}, N_{d,j,k}, N_{d,j}$, and N_d using the new topic label $\bar{z}$ and sentiment label $\bar{k}$;
    end
  end
  for *every 5 iterations* **do**
    Update hyperparameter $\boldsymbol{\alpha}$ with the maximum likelihood estimation according to Eq. 8
  end
  if *number of iterations* = max *Gibbs sampling iterations* **then**
    Update Θ, Π and Φ with new sampling results given by Eqs 7, 6 and 5
  else
    True
  end
end

Notice that the only difference between the generative processes of LJST and Bi-LJST is the way the sampling for topic and sentiment are done for each document. While LJST uses single words from each document to sample topic-sentiment pairs, Bi-LJST uses bi-terms. Due to this subtle difference, Bi-LJST model normally forces the words co-occurring within the same document to be assigned the same topic and sentiment. Further, by using bi-terms, one can employ a larger number of Gibbs sampling iterations to achieve a more robust and coherent topic assignment for each document, as we see in "Experimental Results". However, to our knowledge there is no other body of work which has proposed a theoretical justification as to why bi-terms produce a superior performance over single terms in short texts.

Datasets and Baselines

We use two publicly available datasets—Stanford Treebank (STD) [36] and Yelp review (YRD) [34]. We also use an internal feedback dataset (IFD) (on healthcare applications) gathered by our company (optum global solutions). As our work is primarily intended for short texts, we remove all the reviews that have more than 50 words or fewer than five words. Further, to show the efficacy of our model on particularly shorter texts, we take a subset of YRD with reviews having less than 15 words and create a separate dataset $\text{YRD}_{\text{short}}$. Table 3 shows statistics of all these datasets. The following sections will elaborate more on these datasets for the individual tasks.

We use three joint topic-sentiment methods as baselines: TSM [21], JST [19], and RJST [19]. Further, we use sLDA model to evaluate our model on the basis of sentiment prediction under supervision. We use BTM and LDA models to evaluate the quality of the topics generated using our model. Both BTM and LDA models do not have any associated sentiment layer with them, due to which sentiment prediction accuracy metric is not applicable for these two models.

Hyperparameter Setting

For document-topic distribution, we chose $\boldsymbol{\alpha}$ as the asymmetric prior. For initialization, we empirically chose $\boldsymbol{\alpha} = 10/T$, where T is the number of topics. In Gibbs sampling, after every 5 iterations, we estimate $\boldsymbol{\alpha}$ using Eq. (8). Similar to RJST, we use symmetric $\beta = 0.01$. The Dirichlet parameter γ is the asymmetric prior as described in "LJST: Our Proposed Model". For initialization, we use $\gamma = 10/(T \times S)$. Depending upon the document sentiment label, γ is different for each document. Also for hold out set, as mentioned in "Experimental Results", we use only symmetric γ.

Most of the previous studies defined model priors based on lexicons. Our study shows that lexicons fail to increase model performance in experiments wherein domain-specific keywords carry most of the polarity values. However, in our experiments, we use prior knowledge from MPQA and appraisal lexicons dataset mentioned in Ref. [19] for TSM, JST and RJST. For LJST models, we do not use any lexicon.

In all these methods, Gibbs sampling is run for 1000 iterations. The results reported in the paper are averaged over 50 iterations. In Bi-LJST, instead of capturing all the bigrams from the document, we take only word-pairs using a suitable skip-gram model.

Experimental Results

We use two broad evaluation criteria—sentiment accuracy and topic quality. For sentiment analysis, we split each dataset such that 80% is used for training and remaining 20% for testing. For topic quality evaluation, however, we use the full datasets.

Sentiment Prediction

The reviews in all the datasets are assigned with continuous valued sentiment scores (the range of the score also varies across datasets). As mentioned in "LJST: Our Proposed Model", we scale these scores in [0, 1] range (using min–max normalization), discretize sentiment values into S (number of sentiment labels) equal buckets, and apply joint sentiment-topic models with these discrete sentiment scores.

Table 4 Performance of the models for sentiment prediction and topic extraction

Data	Parameters	Model	Sentiment	Topic	
			RMSE	TSCS	H-score
IFD	T = 10; S = 10	TSM	0.184	−27.630	0.609
		JST	0.231	−28.003	0.546
		RJST	0.211	−23.350	0.499
		sLDA	0.205	−23.302	0.462
		BTM	–	−22.135	0.475
		LDA	–	−24.716	0.610
		LJST	0.186	−22.707	0.553
		Bi-LJST	**0.178**	**−22.085**	**0.314**
STF	T = 10; S = 10	TSM	0.233	−32.406	0.557
		JST	0.245	−40.718	0.563
		RJST	0.239	−36.063	0.447
		sLDA	0.228	−39.639	0.451
		BTM	–	−32.802	0.393
		LDA	–	−37.993	0.599
		LJST	0.241	−36.306	0.423
		Bi-LJST	**0.223**	**−31.804**	**0.144**
YRD	T = 10; S = 5	TSM	0.356	−23.397	0.540
		JST	0.345	−29.343	0.695
		RJST	0.350	−23.484	0.644
		sLDA	0.343	−24.790	0.535
		BTM	–	−21.090	0.425
		LDA	–	−23.920	0.478
		LJST	0.320	−23.400	0.638
		Bi-LJST	**0.313**	**−20.985**	**0.284**
$\text{YRD}_{\text{short}}$	T = 10; S = 5	TSM	0.355	−22.609	0.492
		JST	0.407	−35.409	0.475
		RJST	0.347	−22.264	0.504
		sLDA	0.315	−21.557	0.501
		BTM	–	−23.498	0.408
		LDA	–	−22.553	0.583
		LJST	0.314	−21.991	0.411
		Bi-LJST	**0.311**	**−18.328**	**0.355**

To recalculate our original continuous sentiment scores from discrete classes, we use the following equation:

$$s(d) = \sum_{k=0}^{k=S-1} k \times \text{binsize} \times p(j|d) \times p(k|j,d), \tag{12}$$

where, $p(j|d)$ is the probability of having topic j in document d, $p(k|j, d)$ is the probability of having sentiment label k for topic j in document d, and binsize $= 1/S$. For TSM, we use $p(k|j)$ instead of $p(k|j, d)$ in the last term. For JST, however, we use $p(k|d)$ directly. On the other hand, sLDA treats sentiment analysis as a regression problem. For sLDA, we predict sentiment directly using $p(j|d)$ and the sentiment priors. Notice that we calculate the real-valued sentiment only to evaluate the overall (at a document level) sentiment

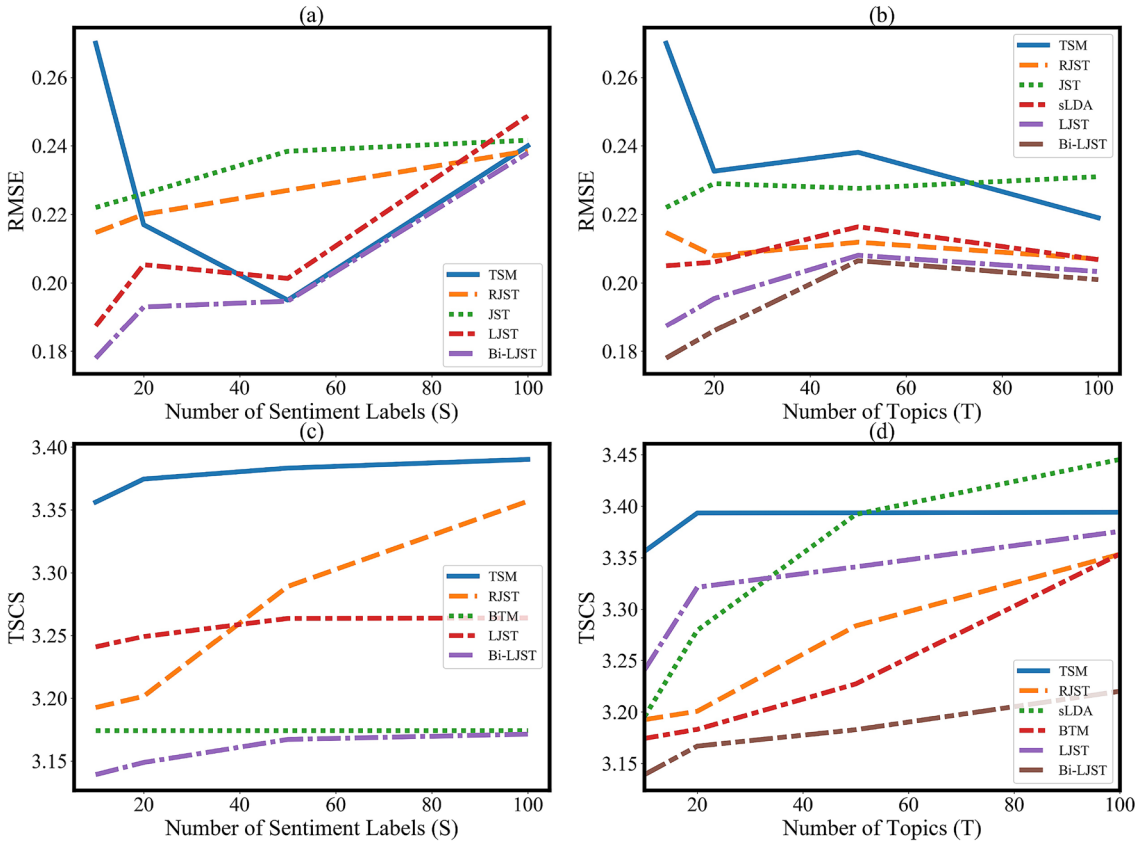


Fig. 2 Performance of the competing models with varying S (keeping $T = 10$) and T (keeping $S = 10$) on IFD dataset for sentiment prediction (**a**, **b**) and topic discovery (**c**, **d**). TSCS value is scaled by $\log(-x)$

prediction task. Although joint topic-sentiment assignments always assign discrete sentiment label to each topic.

We compare the predicted sentiment value with the actual sentiment value on test set using root mean squared error (RMSE). Table 4 shows the RMSE value for all the competing models on three datasets under the best parameter settings. Bi-LJST turns out be the best method by recording RMSE values which are more than 10% lower than RMSE values averaged over all the datasets. The improvement is not so significant in case of STF dataset in comparison to other two datasets, as STF contains more number of texts that are shorter in length, which makes it difficult to capture sentiments properly even with LJST. On the other hand, TSM produces surprisingly better results than JST and RJST methods on IFD and STF datasets, which can be due to the fact that TSM extracts topic-sentiment association at a global level (for the whole corpus). This feature helps TSM to capture document-level sentiment better where topics have predominantly one sentiment polarity, that correlates with overall document sentiment. For example, in IFD most of the sentiment polarities are negative as the texts originate from member grievance channels. It is interesting to notice that sLDA model shows highly competitive results in sentiment prediction task due to the sentiment supervision. Also, given the parametric assumptions of sLDA model it performs surprisingly better than LJST unigram model for short texts. If we see the performance of our models on the datasets YRD and $\text{YRD}_{\text{short}}$, H-score of the latter is much lower than that of the former which indicates the fact that more coherent topics are produced for $\text{YRD}_{\text{short}}$ dataset. This obviously supports the claim that Bi-LJST performs better on short texts.

We also report the performance by varying T and S in Fig. 2a and b. We observe that both our models behave similarly. Further, due to curse of dimensionality the performance of our models deteriorates when we increase the number of sentiment labels. As sLDA does not have any inherent sentiment labels we could only vary T for sLDA model. On the other hand, increasing number of topics (T) tends to have lesser impact on the sentiment prediction.

Topic-specific sentiment prediction: We also undertake the task of evaluating topic-specific sentiment prediction. As the topic-specific ground-truth is not available, we have preferred to build our own dataset. We chose IFD dataset (for it being an internal resource the annotators are familiar with this dataset). We chose the top 5 frequently occurring topics

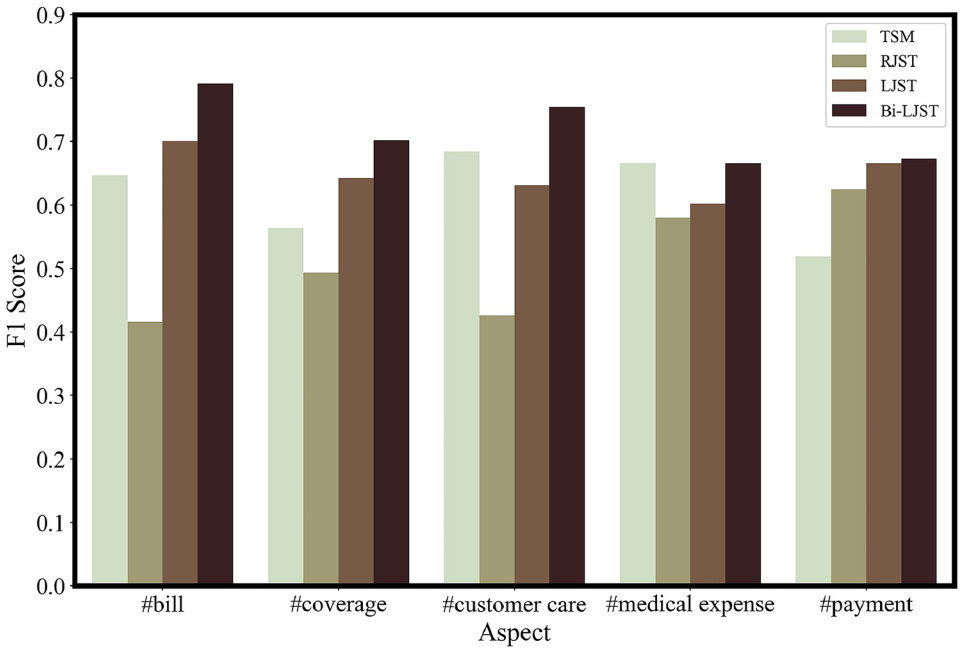


Fig. 3 Performance of the competing methods for topic-specific sentiment classification—a case study on five terms related to "claims payment"

based on the topics extracted by our method, and for each topic treat the top 5 keywords with high marginal probability as the representative items of the topic. Three annotators are then asked to label each feedback-topic pair as positive and negative[2]. The inter-annotator agreement based on Cohen's κ is made to be fixed at 0.82.

We compare LJST and Bi-LJST with TSM and RJST. We cannot compare them with JST here as the latter has the provision of topic being assigned under a given sentiment value, and does not generate topic-specific sentiment score. The overall F1 score averaged over all the topics is as follows: LJST (0.58), Bi-LJST (0.63), TSM (0.57) and RJST (0.53). As a case study, we also choose top five highly probable terms (*a.k.a* aspects) occurring in topic *claims payment*. In all cases, Bi-LJST captures topic-specific sentiment better than other models as shown in Fig. 3. Investigating further into this behavior we realize that our models can capture domain specific phrases such as "better reimburse", "payment denied", "claim coverage" and learn their sentiment polarity based on overall text sentiment and word co-occurrences, whereas baseline models learn sentiment from generic words like "fix", "issue", "better" etc. We observe that for popular aspects such as "customer service", "expenses" and the like, TSM performs similar to LJST and Bi-LJST. Upon analyzing this behavior we find that "customer service" is the most popular aspect in the whole IFD (appears in 12% of the reviews) dataset with skewed sentiment values (more than 80% of the texts are labeled with negative sentiments). In this scenario TSM too performs well as it has a tendency to capture global nature of the corpus. The top words extracted by TSM for the above topic are "concern", "representative", "complaint" etc., which are very generic in nature and are most commonly found phrases in the whole corpus. On the other hand, Bi-LJST (also LJST) extracts phrases such as "deni", "professional", "knowledgable", etc., under the same topic, which are contextually more meaningful and distinctive. In case of rare aspects like "payment" (with 2.4% occurrence rate) LJST extracts phrases like "medicare", "cover", "reimburse". Bi-LJST additionally extracts "paperwork" and "surgery". We report few of such example instances in Table 5. With the help of external sentiment label, LJST and Bi-LJST can learn the connections between domain-specific keywords and their sentiment polarity. For example, a healthcare professional may understand that "paperwork" can be tedious during payments or, medical encounters. However, this knowledge is very difficult to be learnt for any model without prior supervision. In case of LJST (also, Bi-LJST), this supervision can be injected by using user labeled external sentiment. But, other models such as TSM and JST fail to capture this knowledge.

In Table 6 we report the topic-sentiment assignment results of Bi-LJST and other baselines on few instances from IFD. We assign topic labels in $\{1, 2, \ldots, 10\}$ and set discrete sentiment scores on a scale of [1, 10] assigned to each topic. A high sentiment label indicates a positive sentiment polarity

[2] Note that we do not deal with neutral sentiment in this work, mainly because non-subjective feedback is seldom of any importance in case of mining of opinion.

Table 5 Top 5 words under positive and negative sentiment polarities for three topics from IFD

Customer service		Authorization		Claims	
Positive	Negative	Positive	Negative	Positive	Negative
Bi-LJST					
Professional	Dead	Fast	Afford	Policy	Copay
Knowledgeable	Information	Author	Late	Coverage	Payment
Customer	Hang	Copay	Expensive	Payment	Rebut
Efficient	Unavailable	Excellent	Nonpay	Reimbursement	Expensive
Language	Rude	Free	Paperwork	Surgery	Denied
RJST					
Customer	Rude	Authorization	Expensive	Authorization	Payment
Callback	Difficult	Doctor	Website	Prior	Surgery
Excellent	Horrible	Clear	Prior	Network	Waiting
Representative	Hang	Fast	Deliver	Surgery	Approval
Prompt	Waiting	Efficient	Late	Great	Reject
TSM					
Customer	Rude	Efficient	Late	Policy	Network
Great	Complaint	Fast	UHC	Claim	Cost
Excellent	Hang	UHC	Cost	Hospital	Expensive
Timely	Horrible	Great	Expensive	Doctor	Charges
Representative	Pathetic	Perfect	Denied	Helpful	Frustrating

We associate sentiment labels with discrete value of $[8-10]$ as positive and $[1-3]$ as negative

Table 6 Sample texts and the associated sentiment value from IFD

Text	Sentiment	Bi-LJST		RJST		TSM	
		T. L.	S. L.	T. L.	S. L.	T. L.	S. L.
Prior authorizations are very difficult to obtain and having to wait up to 15 business days for a surgical approval is very difficult for both staff and patients	−4.3	1 (0.65)	2 (0.89)	2 (0.39)	3 (0.59)	8 (0.49)	5 (0.55)
The reps at UHC are very knowledgeable to work with	3.9	5 (0.79)	8 (0.91)	3 (0.48)	6 (0.35)	4 (0.68)	4 (0.53)
Your customer service is very difficult to understand and to communicate with	−4.1	5 (0.83)	2 (0.78)	3 (0.62)	3 (0.73)	5 (0.38)	3 (0.61)

We report the T.L. topic label and associated sentiment label S.L. along with the probabilities of assignments (in parenthesis) for Bi-LJST and other baselines. We use $T = 10, S = 10$ for this analysis. A high(low) sentiment label indicates positive(negative) topic-sentiment polarity. Under Bi-LJST model topic id 1 and 5 correspond to claims, and customer service respectively. With RJST modeling topic id 2 and 3 indicate claims and representative respectively. For TSM model, topic id 4, 5 and 8 correspond to topics authorization, representative and customer service respectively

for a topic. We observe that Bi-LJST assigns more coherent topic-sentiment assignments to texts as compared to other baselines. In row 2 in the table, Bi-LJST assigns topic id 5 to the text which corresponds to the topic of customer service. We observe that the sentiment label associated with this topic is 8, indicative of a positive sentiment. Example 3 can be analyzed along the similar lines. Notice that our model is able to distinguish the semantic polarity switch between examples 2 and 3 which can be corroborated by the fact that both the texts are assigned the same topic, but one is labeled with positive sentiment while the other one gets negative sentiment polarity. Another interesting observation can be made on the assignment probabilities computed by each of these models. We observe that Bi-LJST assigns the topmost (associated with highest probability value) topic and the sentiment under this topic with very probability, which shows the confidence on the assignments. On the other hand, the baselines assign labels with lower probabilities, which can be ambiguous in many cases. Although RJST and TSM are able to extract the correct semantic topic for each of the

texts, the sentiment assignments are not always aligned with the overall intent.

Topic Extraction

To evaluate the quality of the topics extracted by the models, we use two quantitative metrics, topic sentiment coherence score (TSCS), and H-score [42]. The former metric is proposed by us for evaluating the quality of topic and sentiment assignment, which is motivated by the ideas of coherence score [24, 42]. Given a topic z, a sentiment label l and its top K words $V^{(z,l)} = \{v_1^{(z,l)}, v_2^{(z,l)}, \ldots, v_K^{(z,l)}\}$, ordered by $p(w|z, l)$, TSCS is defined as below:

$$C(z, l, V^{(z,l)}) = \sum_{j=2}^{j=K} \sum_{i=1}^{i=j} \log \frac{D(v_j^{(z,l)}, v_i^{(z,l)}) + 1}{D(v_i^{(z,l)})}, \tag{13}$$

where $D(v_j^{(z,l)}, v_i^{(z,l)})$ is the number of documents in which $v_j^{(z,l)}$ and $v_i^{(z,l)}$ co-occur together. Similarly, $D(v_i^{(z,l)})$ is the number of documents containing the word $v_i^{(z,l)}$. Finally, the topic sentiment coherence score of a model is calculated by averaging $C(z, l, V^{(z,l)})$ over all z and l. The idea behind TSCS is that words belonging together to a single topical concept will tend to co-occur within the same document. The larger the TSCS value is, the tighter the word pairs are, which in turn, makes topics more coherent and human-interpretable. Table 4 shows that Bi-LJST always outperforms all other baseline models including LJST.

Figure 2c and d shows the change in performance (TSCS) of the models with varying S and T on IFD dataset. Since TSCS value is always negative, we transform the scores to positive values using $\log(-x)$ scaling for better visualization. We notice that increasing the number of topics and sentiment labels adversely affects the performance. This is due to the fact that larger number of topics can produce pronounced similarity between different pairs of topics. However, parameter selection has much lesser impact on Bi-LJST model than any other models, which makes Bi-LJST robust for topic and sentiment identification for short texts.[3]

Next, we discuss about the evaluation of the quality of topics for each document. As topic discovery is very closely related to document clustering, we use H-score based on Jensen–Leibler divergence proposed by [42]. If the average inter-cluster distance is larger than the average intra-cluster distance, i.e., if H-score is low, then the clusters are tightly coupled, which implies that documents sharing similar topic distribution are close to each other. Table 4 shows that Bi-LJST model outperforms any other baselines by a significant margin. It defeats the best baseline by 24% higher H-score.

[3] Similar results are obtained for other datasets which we omit due to lack of space.

Due to strong association among bi-terms, Bi-LJST can identify much richer word-document association than unigram based approaches. Further, the use of sentiment labels of texts helps LJST models to differentiate similar words under different sentiments.

Conclusion

In this paper, we presented a novel framework for joint extraction of sentiment and topics, particularly for short texts. Four major contributions of our work are, Novel Model: our proposed models are informed by the external sentiment labels which in turn reinforce the extraction of better topics and predict better sentiment scores; New Dataset: we created a new real-world dataset (IFD) where the topics are marked with ground-truth sentiment scores; New Metric: we proposed TSCS, a new metric to quantify the topic-sentiment coherence; and Extensive Evaluation: through a rigorous experiment, we showed the superiority of our models. Bi-LJST is deployed and is currently running live.

Appendix

Detailed Formulation of Model Inference for LJST

The total probability of the model based words, topics and sentiment labels can be decomposed as follows:

$$p(\mathbf{w}, \mathbf{z}, \mathbf{l}, \varphi, \pi, \theta, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) = \prod_{j=1}^{T} \prod_{k=1}^{S} p(\varphi_{j,k}; \beta) \times \prod_{d=1}^{D} p(\pi_{d,j}; \boldsymbol{\gamma}) p(\theta_d; \boldsymbol{\alpha}) \times \prod_{i=1}^{N_d} p(z_{d,i}|\theta_d) p(l_{d,j,i}|\pi_{d,j}) p\left(w_{d,j,k,i}|\varphi_{z_{d,i}, l_{d,j,i}}\right). \tag{14}$$

To use collapsed Gibbs sampling, we integrate out φ, π and θ in Eq. (14):

$$\begin{aligned} & p(\mathbf{w}, \mathbf{z}, \mathbf{l}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \\ &= \int_{\varphi} \int_{\pi} \int_{\theta} p(\mathbf{w}, \mathbf{z}, \mathbf{l}, \varphi, \pi, \theta, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \, d\varphi \, d\pi \, d\theta \\ &= \int_{\varphi} \prod_{j} \prod_{k} p(\varphi_{j,k}; \beta) \prod_{d} \prod_{i} p\left(w_{d,j,k,i}|\varphi_{z_{d,i}, l_{d,j,i}}\right) d\varphi \times \\ & \quad \int_{\pi} \prod_{d} \prod_{j} p(\pi_{d,j}; \boldsymbol{\gamma}) \prod_{i} p(l_{d,j,i}|\pi_{d,j}) \, d\pi \times \\ & \quad \int_{\theta} \prod_{d} \prod_{i} p(z_{d,i}|\theta_d) p(\theta_d; \boldsymbol{\alpha}) \, d\theta. \end{aligned} \tag{15}$$

As, φ, π and θ are independent variables, we can split the three terms of the RHS in Eq. (15) and calculate them

separately. Further, each individual terms $\varphi_{j,k}$, $\pi_{d,j}$ and θ_d are independent. Thus, we can interchange the product and integration in each of the three terms.

For each document d, θ_d, we replace the term $p(\theta_d;\boldsymbol{\alpha})$ with corresponding Dirichlet distribution and $p(z_{d,i}|\theta_d)$ with multinomial distribution to get:

$$\int_{\theta_d} p(\theta_d;\boldsymbol{\alpha}) \prod_i p(z_{d,i}|\theta_d)\, d\theta_d = \int_{\theta_d} \left(\frac{\Gamma\left(\sum_{j=1}^{T} \alpha_j\right)}{\prod_{j=1}^{T} \Gamma(\alpha_j)} \right) \theta_{d,j}^{N_{d,j}} \prod_j \theta_{d,j}^{\alpha_j - 1}\, d\theta_d. \tag{16}$$

Using the property of Dirichlet distribution we get:

$$\int_{\theta_d} \left(\frac{\Gamma\left(\sum_{j=1}^{T} \alpha_j + \sum_{j=1}^{T} N_{d,j}\right)}{\prod_{j=1}^{T} \Gamma(\alpha_j + N_{d,j})} \right) \prod_j \theta_{d,j}^{N_{d,j}+\alpha_j - 1}\, d\theta_d = 1.$$

This leads to :

$$\begin{aligned} p(z) &= \prod_d p(z_d) \\ &= \prod_d \int_{\theta_d} p(\theta_d;\boldsymbol{\alpha}) \prod_i p(z_{d,i}|\theta_d)\, d\theta_d \\ &= \left(\frac{\Gamma\left(\sum_{j=1}^{T} \alpha_j\right)}{\prod_{j=1}^{T} \Gamma(\alpha_j)} \right)^D \cdot \prod_d \frac{\prod_j \Gamma(N_{d,j} + \alpha_j)}{\Gamma\left(N_d + \sum_j \alpha_j\right)}. \end{aligned} \tag{17}$$

Similarly, we calculate $p(l) = \prod_d \prod_j p(l_{d,j})$ using the formula below:

$$\begin{aligned} &\prod_d \prod_j \int_{\pi_{d,j}} p(\pi_{d,j};\boldsymbol{\gamma_d}) \prod_i p(l_{d,j,i}|\pi_{d,j})\, d\pi_{d,j} \\ &= \prod_d \prod_j \int_{\pi_{d,j}} \prod_k \pi_{d,j,k}^{\gamma_{d,k}-1} \left(\frac{\Gamma\left(\sum_{k=1}^{S} \gamma_{d,k}\right)}{\prod_{k=1}^{S} \Gamma(\gamma_{d,k})} \right) \pi_{d,j,k}^{N_{d,j,k}}\, d\pi_{d,j} \\ &= \left(\frac{\Gamma\left(\sum_{k=1}^{S} \gamma_{d,k}\right)}{\prod_{k=1}^{S} \Gamma(\gamma_{d,k})} \right)^{D\times T} \cdot \prod_d \prod_j \frac{\prod_k \Gamma(N_{d,j,k} + \gamma_{d,k})}{\Gamma(N_{d,j} + \sum_k \gamma_{d,k})}. \end{aligned} \tag{18}$$

Finally, we obtain $p(w) = \prod_j \prod_k p(w_{j,k})$ by replacing $\prod_{d=1}^{D} \prod_{i=1}^{N_d}$ with $\prod_{i=1}^{V}$ in the formula

$$\begin{aligned} &\prod_j \prod_k \int_{\varphi_{j,k}} p(\varphi_{j,k};\beta) \prod_d \prod_i p(w_{d,j,k,i}|\varphi_{j,k})\, d\varphi_{j,k} \\ &= \prod_j \prod_k \int_{\varphi_{j,k}} p(\varphi_{j,k};\beta) \prod_i p(w_{j,k,i}|\varphi_{j,k})\, d\varphi_{j,k} \\ &= \prod_j \prod_k \int_{\varphi_{j,k}} \prod_i \varphi_{j,k,i}^{\beta-1} \left(\frac{\Gamma(\sum_{i=1}^{V} \beta)}{\prod_{i=1}^{V} \Gamma(\beta)} \right) \varphi_{j,k,i}^{N_{j,k,i}}\, d\varphi_{j,k} \\ &= \left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V} \right)^{T\times S} \cdot \prod_j \prod_k \frac{\prod_i \Gamma(N_{j,k,i} + \beta)}{\Gamma(N_{j,k} + V\beta)}. \end{aligned} \tag{19}$$

The goal of Gibbs sampling is to calculate $p(\mathbf{z}, \mathbf{l} \mid w_t, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma})$ instead of Eq. (15), by taking approximation $p(z_t, l_t \mid w_t, \mathbf{z^{-t}}, \mathbf{l^{-t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma})$, for each word w_t, in document d. Here t is the index of the word in document d.

$$\begin{aligned} &p(z_t, l_t \mid w_t, \mathbf{z^{-t}}, \mathbf{l^{-t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \\ &\quad \propto p(z_t, l_t, \mathbf{z^{-t}}, \mathbf{l^{-t}} \mid w_t, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \\ &\quad = p(z_t, l_t \mid w_t, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \cdot p(, \mathbf{z^{-t}}, \mathbf{l^{-t}} \mid w_t, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}). \end{aligned} \tag{20}$$

If we put the values from Eqs. (17), (18) and (19) in each of the terms in Eq. (20), we get

$$p(z_t, l_t \mid w_t, \mathbf{z^{-t}}, \mathbf{l^{-t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \propto \prod_j \prod_k \frac{\Gamma(N_{j,k,w_t} + \beta)}{\Gamma(N_{j,k} + V\beta)} \cdot \prod_j \frac{\prod_k \Gamma(N_{d,j,k} + \gamma_{d,k})}{\Gamma\left(N_{d,j} + \sum_k \gamma_{d,k}\right)} \cdot \frac{\prod_j \Gamma(N_{d,j} + \alpha_j)}{\Gamma\left(N_d + \sum_j \alpha_j\right)}, \tag{21}$$

and

$$p(z_t = j, l_t = k \mid w_t, \mathbf{z^{-t}}, \mathbf{l^{-t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \propto \frac{\Gamma(N_{j,k,w_t} + \beta)}{\Gamma(N_{j,k} + V\beta)} \cdot \frac{\Gamma(N_{d,j,k} + \gamma_{d,k})}{\Gamma(N_{d,j} + \sum_k \gamma_{d,k})} \cdot \frac{\Gamma(N_{d,j} + \alpha_j)}{\Gamma\left(N_d + \sum_j \alpha_j\right)}. \tag{22}$$

For further simplification, we use formula $\Gamma(x+1) = x \cdot \Gamma(x)$ to get

$$\Gamma(N_{j,k,w_t} + \beta) = \Gamma\left(N^{-t}_{j,k,w_t} + \beta + 1\right) \propto N^{-t}_{j,k,w_t} + \beta.$$

Following this on other terms like $N_{j,k}, N_{d,j,k}, N_{d,j}$ and N_d we get

$$p(z_t = j, l_t = k \mid \mathbf{w}, \mathbf{z^{-t}}, \mathbf{l^{-t}}, \boldsymbol{\alpha}, \beta, \gamma) \propto \frac{N^{-t}_{j,k,w_t} + \beta}{N^{-t}_{j,k} + V\beta} \cdot \frac{N^{-t}_{d,j,k} + \gamma_{d,k}}{N^{-t}_{d,j} + \sum_k \gamma_{d,k}} \cdot \frac{N^{-t}_{d,j} + \alpha_j}{N^{-t}_{d} + \sum_j \alpha_j}. \tag{23}$$

Detailed Formulation of Model Inference for Bi-LJST

There are some changes in the probability distribution for Bi-LJST. Let $\mathcal{B}$ snd B_d denote the vocabulary of bi-terms and the number of bi-terms in the document d. Further assume that a bi-term is represented as $b_t = (w_p, w_q)$. The joint probability of bi-terms, topics and sentiment labels can be captured by,

$$p(b_t, \mathbf{z}, \mathbf{l}) = p(b_t \mid \mathbf{l}, \mathbf{z}) p(\mathbf{l}|\mathbf{z}) p(\mathbf{z}). \tag{24}$$

As the terms, $p(\mathbf{l}|\mathbf{z})$ and $p(\mathbf{z})$ do not depend on bi-term b_t, we can directly make use of Eqs. (17) and (18). Further, in the generative process of Bi-LJST, word pairs within a bi-term are conditionally independent i.e. $p(b_t \mid \mathbf{l}, \mathbf{z}) = p(w_p \mid \mathbf{l}, \mathbf{z}) \cdot p(w_q \mid \mathbf{l}, \mathbf{z})$. This leads us to topic sentiment assignment corresponding to bi-term b_t as follows:

$$p(z_t = j, l_t = k \mid b_t, \mathbf{z}^{-\mathbf{t}}, \mathbf{l}^{-\mathbf{t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \propto \frac{\left(N^{-t}_{j,k,w_p} + \beta\right) \cdot \left(N^{-t}_{j,k,w_q} + \beta\right)}{\left(N^{-t}_{j,k} + V\beta\right)^2} \cdot \frac{N^{-t}_{d,j,k} + \gamma_{d,k}}{N^{-t}_{d,j} + \sum_k \gamma_{d,k}} \cdot \frac{N^{-t}_{d,j} + \alpha_j}{N^{-t}_{d} + \sum_j \alpha_j}, \tag{25}$$

where, p_1 and p_2 are the index of w_p and w_q respectively in vocabulary.

Funding The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Abirami A, Gayathri V. A survey on sentiment analysis methods and approach. In: Eighth international conference on advanced computing (ICoAC), IEEE; 2017. p. 72–76.
2. Ali F, Kwak D, Khan P, El-Sappagh SHA, Ali A, Ullah S, Kim K, Kwak KS. Transportation sentiment analysis using word embedding and ontology-based topic modeling. Knowl Based Syst. 2019;174:27–42.
3. Andreevskaia A, Bergler S. When specialists and generalists work together: overcoming domain dependence in sentiment tagging. In: ACL, Proceedings of the 46th annual meeting of the association for computational linguistics, Columbus, OH, USA; 2008. p. 290–298.
4. Blei DM. Probabilistic topic models. Commun ACM. 2012;55(4):77–84.
5. Blei DM, McAuliffe JD. Supervised topic models. In: Advances in neural information processing systems 20, Proceedings of the twenty-first annual conference on neural information processing systems, Vancouver, BC, Canada; 2007. p. 121–128.
6. Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. J Mach Learn Res. 2003;3:993–1022.
7. Chen Y, Zhang H, Liu R, Ye Z, Lin J. Experimental explorations on short text topic mining between LDA and NMF based schemes. Knowl Based Syst. 2019;163:1–13.
8. Cheng X, Guo J, Liu S, Wang Y, Yan X. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: Proceedings of the 13th SIAM international conference on data mining, Austin, TX, USA; 2013. p. 749–757.
9. Choo J, Lee C, Reddy C, Park H. UTOPIAN: user-driven topic modeling based on interactive nonnegative matrix factorization. IEEE Trans Vis Comput Graph. 2013;19(12):1992–2001.
10. Daud A, Li J, Zhou L, Muhammad F. Knowledge discovery through directed probabilistic topic models: a survey. Front Comput Sci China. 2010;4(2):280–301.
11. Dermouche M, Khouas L, Velcin J, Loudcher S. A joint model for topic-sentiment modeling from text. In: Proceedings of the 30th annual ACM symposium on applied computing (SAC), Salamanca, Spain, ACM; 2015. p. 819–824.
12. Fu X, Yang K, Huang JZ, Cui L. Dynamic non-parametric joint sentiment topic mixture model. Knowl Based Syst. 2015;82(C):102–14.
13. Fu X, Sun X, Wu H, Cui L, Huang JZ. Weakly supervised topic sentiment joint model with word embeddings. Knowl Based Syst. 2018;147:43–54.
14. Gupta D, Singh K, Chakrabarti S, Chakraborty T. Multi-task learning for target-dependent sentiment classification. In: Advances in knowledge discovery and data mining - 23rd Pacific-Asia conference, PAKDD, Macau, China, Proceedings, Part I, Springer, Lecture Notes in Computer Science; 2019. Vol. 11439, p. 185–197.
15. Hofmann T. Probabilistic latent semantic indexing. SIGIR Forum. 2017;51(2):211–8.
16. Hu Y, John A, Wang F, Kambhampati S. ET-LDA: joint topic modeling for aligning events and their twitter feedback. In: Proceedings of the twenty-sixth AAAI conference on artificial intelligence, Toronto, ON, Canada, AAAI Press; 2012.
17. Jelodar H, Wang Y, Yuan C, Feng X, Jiang X, Li Y, Zhao L. Latent Dirichlet allocation (LDA) and Topic modeling: models, applications, a survey. Multimed Tools Appl. 2019;78:15169–15211. https://doi.org/10.1007/s11042-018-6894-4.
18. Kim SM, Hovy E. Determining the sentiment of opinions. In: Proceedings of the 20th international conference on computational linguistics (COLING), Association for Computational Linguistics, USA; 2004. p. 1367–es.
19. Lin C, He Y, Everson R, Rüger SM. Weakly supervised joint sentiment-topic detection from text. IEEE Trans Knowl Data Eng. 2012;24(6):1134–45.
20. Liu B, Zhang L. A survey of opinion mining and sentiment analysis. In: Aggarwal C, Zhai C, editors. Mining text data. Boston: Springer; 2012. p. 415–63.
21. Mei Q, Ling X, Wondra M, Su H, Zhai C. Topic sentiment mixture: modeling facets and opinions in weblogs. In: Proceedings of the 16th international conference on world wide web, WWW, Banff, AB, Canada; 2007. p. 171–180.
22. Mei Q, Shen X, Zhai C. Automatic labeling of multinomial topic models. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining (KDD), San Jose, CA, USA, ACM; 2007. p. 490–499.
23. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. In: 1st International conference on learning representations, ICLR, Scottsdale, AZ, USA, Workshop track proceedings 2013.
24. Mimno DM, Wallach HM, Talley EM, Leenders M, McCallum A. Optimizing semantic coherence in topic models. In: Proceedings of the conference on empirical methods in natural language

processing, EMNLP, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL; 2011. p. 262–272.
25. Minka TP. Estimating a Dirichlet distribution. Tech. Rep. 1, No. 3, Microsoft Research 2000.
26. Nguyen TH, Shirai K. Topic modeling based sentiment analysis on social media for stock market prediction. In: Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing of the Asian Federation of Natural Language Processing, ACL, Beijing, China, vol. 1, Long Papers; 2015. p. 1354–1364.
27. Nugroho R, Yang J, Zhong Y, Paris C, Nepal S. Deriving topics in twitter by exploiting tweet interactions. In: Carminati B, Khan L, editors. IEEE international congress on big data, New York City, NY, IEEE Computer Society; 2015. p. 87–94.
28. Pang B, Lee L. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd annual meeting of the association for computational linguistics, Barcelona, Spain; 2004. p. 271–278.
29. Pang B, Lee L. Opinion mining and sentiment analysis. Found Trends Inf Retr. 2008;2(1–2):1–135.
30. Pang B, Lee L, Vaithyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the conference on empirical methods in natural language processing. Philadelphia, PA, USA: EMNLP; 2002.
31. Rahman MM, Wang H. Hidden topic sentiment model. In: Proceedings of the 25th international conference on world wide web, WWW; 2016. p. 155–165.
32. Ramage D, Hall D, Nallapati R, Manning CD. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In: Proceedings of the conference on empirical methods in natural language processing, EMNLP, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL; 2009. p. 248–256.
33. Roy S, Malladi VV, Gangwar A, Dharmaraj R. A NMF-based learning of topics and clusters for IT maintenance tickets aided by heuristic. In: Information systems in the big data era - CAiSE Forum, Tallinn, Estonia, Proceedings; 2018. p. 209–217.
34. Sabnis O. Yelp reviews dataset. https://www.kaggle.com/omkarsabnis/yelp-reviews-dataset (2018). Accessed 1 Mar 2020.
35. Shi T, Kang K, Choo J, Reddy CK. Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In: Proceedings of the world wide web conference on world wide web, WWW, Lyon, France, ACM; 2018. p. 1105–1114.
36. Socher R, Perelygin A, Wu J, Chuang J, Manning CD, Ng AY, Potts C. Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 conference on empirical methods in natural language processing, EMNLP, Seattle, WA, USA, A meeting of SIGDAT, a Special Interest Group of the ACL; 2013. p. 1631–1642.
37. Supplementary. An anonymized link for code, dataset and appendix. https://github.com/DSRnD/LJST (2019). Accessed 28 Mar 2020.
38. Titov I, McDonald RT. A joint model of text and aspect ratings for sentiment summarization. In: ACL, Proceedings of the 46th annual meeting of the association for computational linguistics, Columbus, OH, USA; 2008. p. 308–316.
39. Turney PD. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting of the association for computational linguistics, Philadelphia, PA, USA; 2002. p. 417–424.
40. Whitelaw C, Garg N, Argamon S. Using appraisal groups for sentiment analysis. In: Herzog O, Schek H, Fuhr N, Chowdhury A, Teiken W (eds) Proceedings of the ACM CIKM international conference on information and knowledge management, Bremen, Germany, ACM; 2005. p. 625–631.
41. Xu W, Liu X, Gong Y. Document clustering based on non-negative matrix factorization. In: SIGIR: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval, Toronto, Canada, ACM; 2003. p. 67–273.
42. Yan X, Guo J, Lan Y, Cheng X. A biterm topic model for short texts. In: 22nd International world wide web conference, WWW. Rio de Janeiro, Brazil; 2013. p. 1445–1456.
43. Zhang L, Wang S, Liu B. Deep learning for sentiment analysis: a survey. Wiley Interdiscip Rev Data Min Knowl Discov. 2018;8(4):e1253.
44. Zhao J, Liu K, Wang G. Adding redundant features for CRFs-based sentence sentiment classification. In: Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, USA, EMNLP; 2008. p 117–126.