An Embedding-based Joint Sentiment-Topic Model for Short Texts

Ayan Sengupta[*†1], **William Scott Paka**[*‡ 2], **Suman Roy**[1], **Gaurav Ranjan**[1] and **Tanmoy Chakraborty**[2]

[1] Optum Global Advantage (OGA), (UnitedHealth Group), Bangalore, India
[2] Dept. of Computer Science & Engineering, IIIT-Delhi, India
{ ayan_sengupta, suman.roy, gauravranjan}@optum.com; {william18026, tanmoy}@iiitd.ac.in

Abstract

Short text is a popular avenue of sharing feedback, opinions and reviews on social media, e-commerce platforms, etc. Many companies need to extract meaningful information (which may include thematic content as well as semantic polarity) out of such short texts to understand users' behaviour. However, obtaining high quality sentiment-associated and human interpretable themes still remains a challenge for short texts. In this paper we develop ELJST, an embedding enhanced generative joint sentiment-topic model that can discover more coherent and diverse topics from short texts. It uses Markov Random Field Regularizer that can be seen as generalisation of skip-gram based models. Further, it can leverage higher order semantic information appearing in word embedding, such as self-attention weights in graphical models. Our results show an average improvement of 10% in topic coherence and 5% in topic diversification over baselines. Finally, ELJST helps understand users' behaviour at more granular levels which can be explained. All these can bring significant values to service and healthcare industries often dealing with customers.

Introduction

Short text is a popular mean of communication in online social media and e-commerce websites that appear abundant in different applications. Mining short texts is thus essential to extract thematic content of the text as well as to identify the sentiment expressed by the customers about certain entities (products, services, and movies to name a few). In many applications it may be required to discover both topic and sentiment simultaneously as seen in target dependent (or topic-specific) sentiment analysis (Gupta et al. 2019).

A Motivation for this work: There have been a few attempts to predict both sentiment and topics simultaneously (Mei, Shen, and Zhai 2007; Lin et al. 2012; Rahman and Wang 2016; Nguyen and Shirai 2015); among which extraction of Joint Sentiment-Topic (JST) model is quite popular. Let us illustrate the functionality of JST compared to ELJST (our proposed method to be introduced subsequently) through the following example of a review:

Claims payment was fast and easy. However, *language barrier* with *customer care* was really *difficult* to deal with. ★★★★★

JST:
claims 👍 0.8 | pay 👍 0.8 | customer 👎 0.45

JST with skip-gram:
claims pay 👍 0.7 | customer care difficult 👎 0.6

ELJST:
claims pay fast 👍 0.85 | customer language barrier 👎 0.7

JST will discover the topics such as `claims`, `pay`, `customer`, etc. Also using skip-gram-based JST model (a skip-gram JST can be developed by assuming a topic distribution over n-grams) one will be able to discover topics like `claims pay`, `customer care difficult` etc. Through the use of an appropriate sentiment lexicon, JST will also detect sentiment values of the topics as shown above, without considering any external sentiment labels (star rating). However, JST suffers from few drawbacks such as using only unlabeled data, for which it is unable to incorporate external labels like the ratings given by the customers or, ground-truth labels obtained from the annotators etc. We believe that external labels often play an important role in determining 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 of the topics. Also JST does not allow context-based information to be used for model discovery, which otherwise using skip-gram model, may lead to better topic quality as we will see later in this paper.

To alleviate these issues, we introduce **Embedding Enhanced Labeled Joint Sentiment Topic (ELJST) Model**, a novel framework that jointly discovers topics and sentiment for short texts in presence of labeled (with discrete sentiment values) texts. ELJST model bears close resemblance to the work of weakly supervised joint topic-sentiment model (Lin et al. 2012), which is an extension of the classic topic model based on Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003). ELJST constructs an extra sentiment layer on top of LDA with the assumption that sentiments are generated based on topic distributions, and words are generated

[*]Equal Contribution.
[†]Corresponding Author.
[‡]This work was done when the author was an intern at OGA during May-July 2019.

by conditioning on the topic-sentiment pairs. To skirt the sparsity problem we do not take recourse to usual practice of topic modeling on short texts using skip-gram (Shi et al. 2018) or, bi-term (Yan et al. 2013) as these models are inherently required to be parameterized with the window size of the context or the length n of n-grams. Rather we use Markov Random Field (MRF) Regularizer that creates an undirected graph for each text by constructing edges between contextually and semantically similar words, and formulates a well-defined potential function to enhance topic identification. In an earlier work (Sengupta, Ranjan, and Roy 2021), we have proposed LJST (Labeled Joint Sentiment-Topic) model particularly for short texts, which also extracts topics and detects sentiment values for documents as well as extracted topics. It uses a probabilistic framework based on Latent Dirichlet Allocation. 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. Thus LJST is a precursor of ELJST in the sense that the latter uses MRF regularizer instead of bi-terms (we define *bi-terms* as a pair of words appearing in any order in the same sentence) used in LJST which results in ELJST performing better than JST in terms of the quality of extracted topics and associated sentiment assignments.

Contributions of our work:

- Our model can generalise skip-gram or n-gram based joint topic-sentiment models by considering long-term dependency between tokens by leveraging embeddings or, self attentions.
- Further, by using overall text labels in a supervised manner, we can avoid using external lexicons and incorporate richer domain specific external knowledge into generative models.
- Our model produces tighter and more coherent latent representation by employing MRF regularizer, which makes topic models to be more human interpretable for short texts.

Related Work

We briefly describe prior art in two parts, that of related to – (1) Joint Sentiment-topic Extraction, and (2) Word Embedding Assisted Topic Extraction and Sentiment Modeling.

Due to the abundance of literature on sentiment analysis and topic modeling, we restrict to studies which we deem pertinent to our work. **Joint Sentiment-Topic Extraction:** Topic-sentiment Model (TSM) (Mei, Shen, and Zhai 2007) is the first attempt to deal with the extraction of sentiment and topic models jointly. As TSM is primarily based on probabilistic latent semantic indexing (pLSI) (Hofmann 1999), it suffers from two common drawbacks: inferring quality topics for new document and over-fitting. To overcome these, (Lin et al. 2012) propose a weakly supervised hierarchical Bayesian model, *viz.* JST. The extraction of JST model ensures topic generation to be conditioned on sentiment labels. The same authors introduce another model, called Reverse-JST (RJST) in which sentiment generation depends on topic.

In (Sengupta, Ranjan, and Roy 2021) Sengupta *et al.* 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 JST model (Lin et al. 2012) wherein we discover topics and predict sentiment for short texts by drawing bi-terms, instead of unigrams according to a topic–bi-term distribution in presence of a collection of labeled short texts.

Sentiment-LDA (sLDA) (Li, Huang, and Zhu 2010) and Dependency-Sentiment-LDA (dsLDA) (Li, Huang, and Zhu 2010) use external sentiment lexicon under global and local context to be linked with topic identification from texts. The labeled topic model proposed in (Ramage et al. 2009) uses external labels for capturing linear projection on the priors. However, all these models are unable to 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 (Rahman and Wang 2016). (Poddar, Hsu, and Lee 2017) propose SURF that identifies opinions expressed in a review. (Nguyen and Shirai 2015) introduce Topic Sentiment Latent Dirichlet Allocation (TSLDA), a new topic model that can capture the topic and sentiment simultaneously. One of the popular topic models employed to automatically extract topical contents from the documents is based on non-negative matrix factorization (NMF) *e.g.*, (Lee and Seung 1999; Xu, Liu, and Gong 2003). However, this usually does not produce sentiment labels. For this, one has to address sentiment prediction problem for short texts (some of which are labeled with discrete or real numbers) using a semi-supervised approach of extracting joint sentiment/topic model. Such a method has been proposed in (Li, Zhang, and Sindhwani 2009) using a constrained non-negative tri-factorization of the term-document matrix implemented using novel yet simple update rules; however, it uses discrete sentiment values only. The authors extended this approach to incorporate real sentiment values lying in a particular range (Roy et al. 2018). But this cannot match up to the accurate sentiment values predicted by other methods for which we do not consider this work for benchmarking purpose.

Word Embedding Assisted Topic Extraction and Sentiment Modeling: Recently, researchers have started using richer word representations to fine-tune topic models in order to extracting more meaningful topics (Qiang et al. 2017; Fu et al. 2018). As mentioned in (Qiang et al. 2017), an embedding-assisted topic model can understand the latent semantic relationship between two words "king" and "queen" and place them under the same topic, irrespective of whether they co-occur in same text or not. Another advantage of using word embedding in topic models lies in its ability to generalize the model. The authors in (Yan et al. 2013) use bigrams instead of unigram in order to tackle sparsity in short

Model	Input data	Lexicon needed?	Word emb. used?	Output		
				T-S	T dist. over W	T polarity
JST (Lin et al. 2012)	UL	Yes	No	T under S	Global	Doc-level
TSM (Mei, Shen, and Zhai 2007)	UL	Yes	No	T-S pair	Local	Global
RJST (Lin et al. 2012)	UL	Yes	No	T-S pair	Local	Doc-level
WS-TSWE (Fu et al. 2018)	UL	No	Yes	T-S pair	Local	Doc-level
LJST (Sengupta, Ranjan, and Roy 2021)	L	No	No	T-S pair	Local	Doc-level
ELJST	L	No	Yes	T-S pair	Local	Doc-level

Table 1: Comparison of ELJST and other baseline methods w.r.t different dimensions (T: Topic, S: Sentiment, W: Word, Doc: Document, UL: Unlabeled, L: Labeled).

texts. Recently in (Fu et al. 2018; Fu, Wu, and Cui 2016), the authors propose a novel topic sentiment joint model called weakly supervised topic sentiment joint model with word embedding (WS-TSWE), which incorporates word embedding and HowNet lexicon simultaneously to improve the topic identification and sentiment recognition. A generalized model is introduced in (Ali et al. 2019) which is able to use n-grams to capture long term dependencies between words. The work on joint Sentiment Topic model aims to deal with the problem about the mixture of topics and sentiment simultaneously. Most of them have gone to show that embedding and joint sentiment-topic joint model can be combined effectively to discover the mixture of topics and sentiment simultaneously.

Table 1 summarizes a comparison of ELJST with existing models w.r.t. different dimensions of the model.

Embedding Enhanced Labeled Joint Topic-Sentiment Model

In this section, we discuss the proposed Embedding enhanced Labeled Joint Topic-Sentiment model (ELJST) for identifying coherent and diverse topics along with sentiment classes extracted from labeled text data.

Our Proposed Model

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. We assume each document d to be labeled with a number $\lambda^d \in \{1, 2, \ldots S\}$. This allows to define a document-specific label projection vector $\mathbf{L}^{(d)}$ of dimension S as:

$$L_k^{(d)} = \begin{cases} 1 & \text{if } \lambda^{\mathrm{d}} = k \\ 0 & \text{otherwise} \end{cases}$$

In other words, the kth entry of $\mathbf{L}^d$ is 1 if the label of document d is k.

We approximate it by $L^{(d)} \leftarrow L^{(d)} + \epsilon$, $0 < \epsilon < 1$. In ELJST model, we say two words are semantically similar if their distance is less than a threshold value. We create an undirected graph G (MRF) for each document d by connecting semantically similar words and their corresponding topic assignments. We identify semantic similarity between two words using appropriate distance metric on various embedding representations, such as Word2Vec (Mikolov et al. 2013; Goldberg and Levy 2014), sub-word level representation (fastText) (Bojanowski et al. 2017; Li et al. 2018), contextual embedding (BERT) (Devlin et al. 2019) and attention models (Bahdanau, Cho, and Bengio 2015; Vaswani et al. 2017). Each of these techniques have their own merits and demerits. Word2Vec is easy to use, although, word embedding for domain specific words are not always available on Word2Vec. fastText (Bojanowski et al. 2017) uses sub-word level representations and can generate word vectors for out of vocabulary words. However, both Word2Vec and fastText produce static embeddings.

On the other hand, BERT (Devlin et al. 2019) can capture contextual information which allows one to construct dynamic edges for same word token in different contexts.

Generative Model of ELJST

Generating a word w_i in document d is a three-stage procedure, as shown in Figure 1. First, a topic j is chosen from 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 ELJST shown in Figure 1 are formalised as below:

Here, $\boldsymbol{\alpha}$ and β are hyperparameters – the former is the prior observation count, denoting the number of times topic j is associated with document d, and the latter is the number of times words sampled from topic j which are associated with sentiment label l before observing the actual words. $\mathrm{Dir}(\cdot)$ is the Dirichlet distribution. 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 (Ramage et al. 2009):

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

The perturbation parameter ϵ is used to forcibly assign non-

1. For each document d
Generate $\theta_d \sim \text{Dir}(\boldsymbol{\alpha})$;
2. For each document d and
topic $j \in \{1, 2, \dots T\}$
Choose $\pi_{d,j} \sim \text{Dir}(\boldsymbol{\gamma}^{(d)})$, $\boldsymbol{\gamma}^{(d)} = \gamma \times \mathbf{L}^{(d)}$;
3. For each topic $j \in \{1, 2, \dots T\}$ and
sentiment label $l \in \{1, 2, \dots S\}$
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 label l_i and
topic z_i.

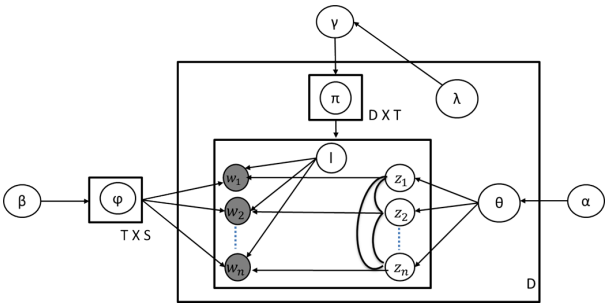


Figure 1: Generative Model of ELJST

zero values to labels. We have used $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ as the asymmetric priors and β as the symmetric prior. 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.

Model Inference and Parameter Estimation

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

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

The first term of Eq. 1 is obtained by integrating w.r.t. φ shown in Eq. 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.

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

The second term of Eq. 1 is obtained by integrating w.r.t. π shown in Eq. 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.

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

We write the third term of Eq. 1 by integrating w.r.t. θ, as shown in Eq. 4, where N_d is the total number of words in document d. As discussed in (Qiang et al. 2017), MRF model defines the binary potential (weight of undirected edge) for each edge (z_{w_i}, z_{w_j}) of undirected graph G_d as $\exp(\mathbb{1}_{z_{w_i}=z_{w_j}})$, where $\mathbb{1}$ is the indicator function. P_d is the set of edges and $|P_d|$ is the total number of edges in undirected graph G_d for dth document. η is an user-specified parameter that controls the effects of MRF Regularization into our model. If $\eta = 0$, then we do not consider the effect of MRF into our model.

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.

$$p(\mathbf{z}) = \left(\frac{\Gamma(\sum_{j=1}^{T} \alpha_j)}{\prod_{j=1}^{T} \Gamma(\alpha_j)} \right)^{D} \cdot \prod_d \frac{\prod_j \Gamma(N_{d,j} + \alpha_j)}{\Gamma(N_d + \sum_{j=1}^{T} \alpha_j)} \cdot \exp(\eta \frac{\sum_{(z_{w_a}, z_{w_b}) \in P_d} \sum_j \mathbb{1}_{z_{w_a}=z_{w_b}}}{|P_d|}) \tag{4}$$

$$p(z_t = j, l_t = k \,|\, w_t, \mathbf{z^{-t}}, \mathbf{l^{-t}}, \boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}) \propto \frac{N^{-t}_{j,k,w_t} + \beta}{N^{-t}_{j,k} + V\beta} \cdot \frac{N^{-t}_{d,j,k} + \gamma_{d,k}}{N^{-t}_{d,j} + \sum_k \gamma_{d,k}} \cdot \frac{N^{-t}_{d,j} + \alpha_j}{N^{-t}_{d} + \sum_j \alpha_j} \cdot \exp(\eta \frac{\sum_{i \in N_{d_{w_t}}} \sum \mathbb{1}_{z_i=j}}{|N_{d_{w_t}}|}) \tag{5}$$

Above $N_{d_{w_t}}$ denotes the words appearing in the document d that are labeled to be similar to word w_t based on the embedding. Similarly, $|N_{d_{w_t}}|$ is the total number of such words.

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

$$\varphi_{j,k,i} = \frac{N_{j,k,i} + \beta}{N_{j,k} + V\beta} \tag{6}$$

The per-document topic specific sentiment distribution is approximately computed as,

$$\pi_{d,j,k} = \frac{N_{d,j,k} + \gamma_{d,k}}{N_{d,j} + \sum_k \gamma_{d,k}} \tag{7}$$

Finally, we approximate per-document topic distribution as,

$$\theta_{d,j} = \frac{N_{d,j} + \alpha_j}{N_d + \sum_j \alpha_j} \tag{8}$$

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

Algorithm 1: Gibbs sampling procedure for ELJST

Input : $\boldsymbol{\alpha}, \beta, \boldsymbol{\gamma}^{(d)}$
Initialization : Initialize matrix $\boldsymbol{\Theta}_{D\times T}$, tensor $\boldsymbol{\Pi}_{D\times T\times S}$, tensor $\Phi_{T\times S\times V}$;

for *i = 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. 5;
      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
  if *number of iterations* = max *Gibbs sampling iterations* **then**
    Update Θ, Π and Φ with new sampling results given by Eqs 8, 7 and 6
  else
    True
  end
end

Experimental Setup

In this section, we describe the setup for the experiments we have performed to demonstrate the effectiveness and robustness of our model over other baselines on 5 datasets.

Baseline Methods

We compare the performance of our model with five baselines mentioned below:

- Dependency-Sentiment-LDA (dsLDA) (Li, Huang, and Zhu 2010), RJST (Lin et al. 2012) and TSM (Mei, Shen, and Zhai 2007), which are traditional LDA based joint topic-sentiment models.
- ETM (Qiang et al. 2017), a short text topic model with word embedding. As ETM does not have sentiment associated, to make results comparable, we use $T \times S$ number of topics.
- WS-TSWE (Fu et al. 2018; Fu, Wu, and Cui 2016) a weakly supervised joint topic-sentiment model with word embedding.
- LJST (Sengupta, Ranjan, and Roy 2021) a semi-supervised joint topic-sentiment model without word embedding. LJST uses labeled and unlabeled data to extract topics and associated sentiment probabilities for each text.

Hyper-Parameter Settings

For document-topic distribution, we chose $\boldsymbol{\alpha}$ as the asymmetric prior. For initialisation, we empirically chose $\boldsymbol{\alpha} = 10/T$, where T is the number of topics. Similar to RJST, we use symmetric $\beta = 0.01$. The Dirichlet parameter γ is the asymmetric prior as described earlier. For initialisation, we use $\gamma = 10/(T \times S)$. Depending upon the document sentiment label, γ is different for each document. Also for test set, as mentioned in the previous section, we use only symmetric γ. For all the methods, same values for $\boldsymbol{\alpha}, \beta$ and γ are used. As suggested in (Qiang et al. 2017), we use $\eta = 1$ for both ETM and ELJST. For WS-TSWE we use $\lambda = 0.1$ and $\mu = 0.01$. In all the methods, Gibbs sampling is run for 1000 iterations. The results reported in the paper are averaged over 5 runs.

MRF Creation in ELJST

We construct a Markov Random Field by connecting semantically similar words with edges for each document. Words in a document are represented as vectors using a suitable word embedding. Recall from previous sections that two words are semantically similar if distance between two word vectors using an appropriate distance metric is less than a threshold value (ε). [1] For representing words we use static word embedding - Word2Vec [2], and sub-word level embedding - fastText [3]. For base Word2Vec (without fine-tuning) we use 300-dimensional word embeddings trained on Google news data [4].

In fine-tuning, we use 300-dimensional embeddings learned on each of the datasets separately. Word2Vec fine-tuning is done using Gensim with default parameter configuration for 20 epochs. Similarly, for fastText we use 300-dimensional embeddings trained on Common Crawl dataset [5]. In fine-tuning, we use 300-dimensional word embeddings learned on each dataset separately. fastText is fine-tuned using fastText library [6] with skip-gram for 20 epochs with a learning rate of 0.5. For unknown vocabulary words we use 300D vectors randomly sampled from glorot-uniform distribution.

We further use the BERT base model[7] fine-tuned on our labelled dataset in the downstream classification task and extract the 768-dimensional vector representation for each word token. For BERT fine-tuning, we use Huggingface's *BertForSequenceClassification* wrapper[8] for sentiment classification task. We use the original pretrained BERT wordpiece tokenizer to tokenize our dataset.[9] The classification model is trained on each of training datasets. We use Adam optimizer with a learning rate of $5e-5$ for 20 epochs, with a early stopping of 5 rounds on the validation dataset. We extract the 768-dimensional word token embeddings and multi-headed self-attention weights between each token pair from the base

[1] threshold value ε is different from the perturbation value ϵ.
[2] https://radimrehurek.com/gensim/models/word2vec.html
[3] https://fasttext.cc/
[4] https://code.google.com/archive/p/word2vec/
[5] https://dl.fbaipublicfiles.com/fasttext/vectors-english/crawl-300d-2M-subword.zip
[6] https://fasttext.cc/docs/en/unsupervised-tutorial.html
[7] https://huggingface.co/transformers/model_doc/bert.html
[8] https://huggingface.co/transformers/model_doc/bert.html\#bertforsequenceclassification
[9] https://huggingface.co/transformers/model_doc/bert.html\#berttokenizer

Embedding	Tuned	ε	Kindle		Movies		Home		IFD		Twitter	
			Mean	Max.	Mean	Max.	Mean	Max.	Mean	Max.	Mean	Max.
Word2Vec	No	0.3	25	258	24	209	23	204	7	31	5	24
Word2Vec	Yes	0.3	28	312	31	239	29	276	9	35	6	27
Word2Vec	No	0.9	3	19	3	18	3	19	3	7	2	5
Word2Vec	Yes	0.9	4	21	5	20	5	27	5	10	2	6
fastText	No	0.3	77	595	84	595	86	712	11	47	6	27
fastText	Yes	0.3	83	617	84	601	87	719	13	53	7	29
fastText	No	0.9	2	16	2	32	2	19	2	13	2	12
fastText	Yes	0.9	2	16	2	33	3	21	2	14	2	15
BERT	Yes	0.9	41	820	48	861	37	816	15	78	11	43
BERT Attention	Yes	NA	71	284	76	291	82	327	8	47	10	39

Table 3: Statistics of constructed number of edges (for each document) for different types of embedding

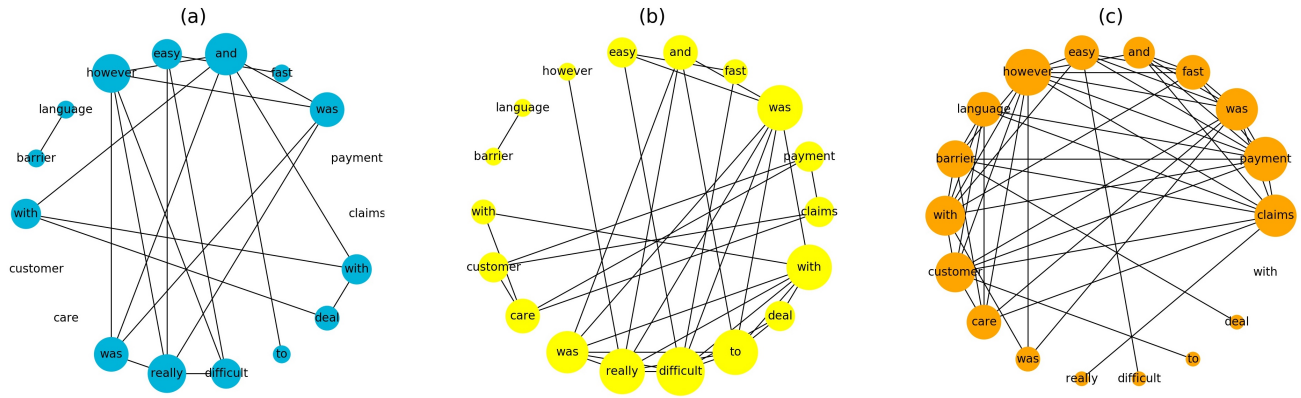


Figure 2: MRF creation for a sample document using different methods: (a) fastText (pre-trained) (b) BERT embedding (c) BERT attention. For fastText we use a threshold of $\varepsilon = 0.3$ and for BERT $\varepsilon = 0.9$. For BERT attention we choose only the token with highest attention value for each word.

BERT model of the fine-tuned classification model. For all the embedding models, cosine similarity is used to measure the similarity between two word vectors. We vary the threshold between 0.3 and 0.9 to observe how the model changes with respect to loosely or densely connected Markov Fields. Best results are observed at $\varepsilon = 0.3$ for most of the embedding models.

BERT-based model consists of 12 layers with 12 self-attention heads each. The attention heads operate in parallel and help the model capture wider range of relationships across words. We first considered all 12 heads from only the last layer. For each of the 12 attention heads, we pick the token with highest attention weight for each word; using this approach we observed that the learning at each attention head is different. Hence, we combine all the 12 heads by taking average to construct the undirected graph for each document. Two words within a document are connected by an edge, if and only if, they attend each other under any attention head. Therefore, mathematically, we construct $e_{ij} = (w_i, w_j)$ between words w_i and w_j from any document d, if and only if

$$j = \arg\max_k \{Attention^{head_n}(w_i, w_k); \text{ for some } n \in \{1, 2, \ldots, 12\}\} \quad (9)$$

$$i = \arg\max_k \{Attention^{head_n}(w_j, w_k); \text{ for some } n \in \{1, 2, \ldots, 12\}\} \quad (10)$$

Note that a few edges from attention could form self loops which will not be considered here.

Table 3 shows statistics of different embedding representations. The mean and average number of edges with a threshold 0.9 is quite low for almost all the models. Threshold 0.3 provides more edges for Word2Vec and fastText. fastText when fine tuned, generates word vectors that are more domain specific and in turn carries more information when computing similarities across words. BERT produce word vectors with high contextual information due to which the similarities are much closer and the threshold is chosen empirically. BERT generates too many edges for few documents making the maximum number of edges go much higher. BERT attention provides appropriate amount of edges where the mean number of edges is not too low and the maximum number of edges is not too high. BERT attention is observed to preserve local as well as global contexts much better than other variations.

Data	Model	Topic			Perp.
		H-Sc.	TSCS	Div.	
Home	dsLDA	0.362	0.174	0.410	5531.9
	ETM	0.193	0.192	**0.770**	5717.4
	RJST	0.360	0.131	0.620	5408.0
	TSM	0.445	0.145	0.540	5966.5
	WS-TSWE	0.253	0.203	0.710	5102.3
	LJST	0.208	0.183	0.670	5016.3
	ELJST ($\eta = 0$)	0.329	0.141	0.600	5201.7
	ELJST	**0.118**	**0.214**	0.740	**4957.2**
Kindle	dsLDA	0.482	0.067	0.220	7643.2
	ETM	0.200	0.182	0.650	6984.0
	RJST	0.387	0.114	0.600	7967.3
	TSM	0.477	0.134	0.560	7966.5
	WS-TSWE	0.176	0.180	0.630	6766.4
	LJST	0.159	0.132	0.700	6819.4
	ELJST ($\eta = 0$)	0.201	0.097	0.450	7014.5
	ELJST	**0.113**	**0.196**	**0.710**	**6513.3**
Movies	dsLDA	0.488	0.166	0.380	5552.1
	ETM	0.178	0.187	0.720	4467.6
	RJST	0.367	0.090	0.630	5842.3
	TSM	0.462	0.125	0.480	5991.7
	WS-TSWE	0.445	0.194	0.690	4008.0
	LJST	0.217	0.209	0.680	4089.1
	ELJST ($\eta = 0$)	0.337	0.112	0.710	4590.1
	ELJST	**0.124**	**0.227**	**0.750**	**3834.7**
IFD	dsLDA	0.613	0.052	0.680	817.25
	ETM	0.431	0.117	0.730	701.03
	RJST	0.558	0.079	0.690	830.11
	TSM	0.542	0.080	0.650	832.55
	WS-TSWE	0.408	0.102	0.740	692.67
	LJST	0.458	0.092	0.600	830.76
	ELJST ($\eta = 0$)	0.529	0.067	0.630	798.06
	ELJST	**0.301**	**0.126**	**0.740**	**681.09**
Twitter	dsLDA	0.511	0.057	0.288	1457.2
	ETM	0.157	0.146	0.300	2208.4
	RJST	0.498	0.198	0.336	1434.3
	TSM	0.492	0.112	0.264	2033.3
	WS-TSWE	0.224	0.186	0.144	1012.8
	LJST	0.101	0.177	0.390	513.82
	ELJST ($\eta = 0$)	0.082	0.173	0.350	279.63
	ELJST	**0.078**	**0.201**	**0.440**	**280.75**

Table 4: Performance of ELJST (with BERT attention) against the baselines. Best performance for all models are observed for Amazon and Twitter datasets at $T = 5$ and for IFD at $T = 10$ (where T = no of topics). **H-Sc.** stands for H-Score, **Div.** stands for Diversity score, **Perp.** stands for Perplexity.

In Figure 2 we show how different embedding methods help extracting different levels of knowledge from texts. A naive version of fastText embedding fails to capture semantic similarities between domain specific words - 'customer', 'care', 'language' and 'barrier', which is realised by other methods. Using BERT attentions we can capture the long term dependencies [10] between words 'claims', 'payment' and 'easy' and similarly between 'customer', 'language' and 'barrier'.

Experimental Results

We evaluate our results in a two-pronged manner, qualitative and quantitative.

[10] one word appearing not in near neighbourhood of another word

Quantitative Evaluation

Our quantitative evaluation is based on measuring the (1) quality of topic sentiment model, (2) quality of topical representation of documents, and (3) quality of document modeling.

To measure the quality of extracted topic sentiment model we use the coherence metric, **Topic-Sentiment Coherence Score** (Dieng, Ruiz, and Blei 2020) (TSCS), which is defined as the average pointwise mutual information of word pairs under each topic-sentiment pair. The larger the TSCS value is, the tighter the word pairs are, which in turn makes topics more coherent and interpretable.

To measure the quality of topical representation we use **Diversity score** (Dieng, Ruiz, and Blei 2020) which helps understand the uniqueness of words generated per topic. A Diversity score close to 0 indicates redundant topics, whereas diversity close to 1 reflects more varied topics.

As topic modeling is closely related to document clustering we use another topical representation quality metric **H-score** (Yan et al. 2013) based on Jensen-Leibler divergence between two documents. A low H-score implies that the average inter-cluster distance is larger than the average intra-cluster distance which results in tightly coupled clusters, hence the documents that share similar topic distribution are close to each other.

To evaluate the generative behaviour of our model we compute the **Perplexity** (Blei, Ng, and Jordan 2003) of the test set. The lower the perplexity, the better is the generative performance of the model.

Table 4 shows the comparison of ELJST to the baseline methods on all the datasets. Among all embedding configurations, the best performance for ELJST is observed under BERT attention settings. For ETM and WS-TSWE however, the best results are observed with fastText fine-tuned embeddings. It is easy to see that ELJST consistently outperforms other baseline methods under all the evaluation metrics. JST based models such as dsLDA, RJST and TSM behave similarly as they are built on similar generative structure.

On the other hand, both ETM and WS-TSWE perform much better in terms of topic quality, as they incorporate contextual information into the models through word embedding. The topic-sentiment pairs identified by ELJST are at least 8% more coherent than the ones extracted by WS-TSWE and ETM models. On the other hand, we observe relatively low variability in the topic diversity, although ELJST demonstrates the highest diversity among all the models.

In document clustering task, ELJST demonstrates a drastic improvement of over 20% over other baselines. Even on shorter texts in IFD and Twitter datasets, ELJST observes more than 30% improvement in the document clustering and more than 1.5% improvement in the topic coherence. In order to show the performance gain due to the utilization of labelled data, we also compare our method with the version with no MRF by setting $\eta = 0$. Table 4 shows that ELJST with MRF always outperforms ELJST with $\eta = 0$. Even in most of the cases, the unsupervised baselines outperform no MRF version of ELJST. This shows the contribution of embeddings into our model. Hence, the superiority of our model is not just due to external labels, rather, due to underlying generative model

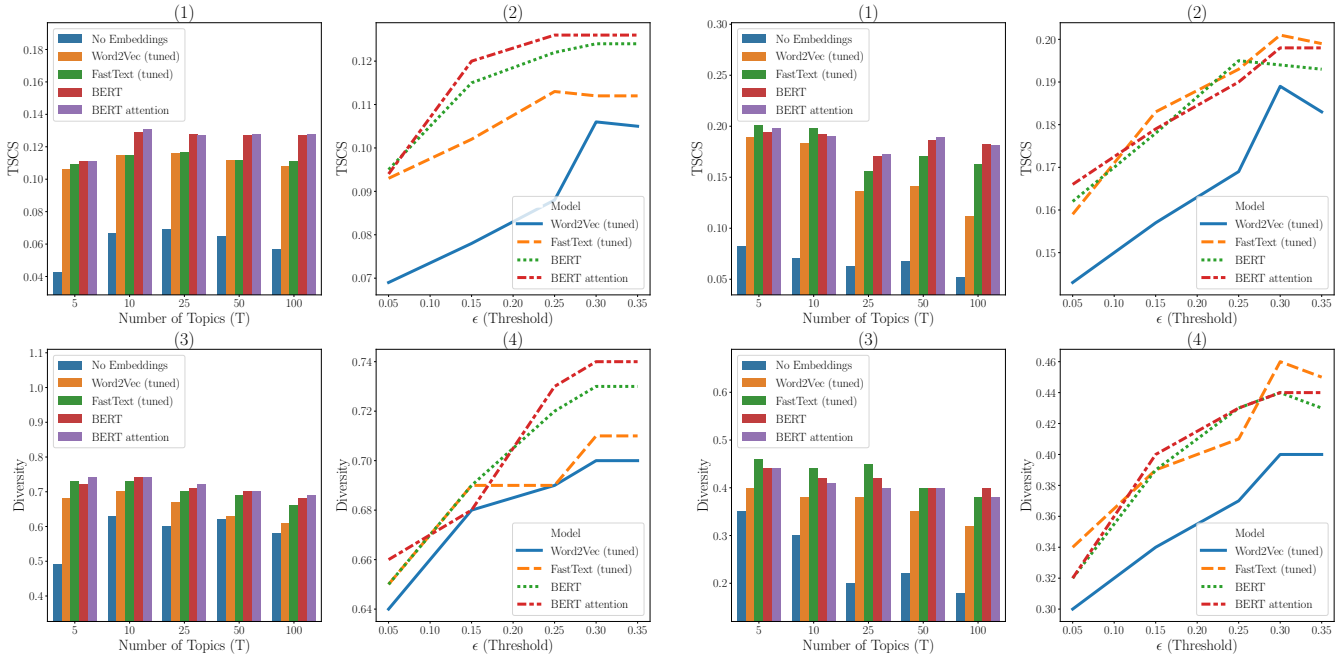
Emb.	Tuned	H-Sc.	TSCS	Div.	Perp.
None ($\eta = 0$)	-	0.529	0.067	0.630	798.06
Word2Vec	No	0.411	0.106	0.700	701.24
Word2Vec	Yes	0.401	0.115	0.710	698.17
fastText	No	0.402	0.113	0.710	693.28
fastText	Yes	0.341	0.124	0.730	690.44
BERT	Yes	0.312	0.125	**0.740**	685.75
BERT Attn.	Yes	**0.301**	**0.126**	**0.740**	**681.09**

(a) IFD Dataset

Emb.	Tuned	H-Sc.	TSCS	Div.	Perp.
None ($\eta = 0$)	-	0.082	0.173	0.350	279.63
Word2Vec	No	0.087	0.189	0.380	277.99
Word2Vec	Yes	0.078	0.180	0.400	263.03
fastText	No	**0.074**	0.190	0.440	277.25
fastText	Yes	0.082	**0.201**	**0.460**	**242.49**
BERT	Yes	0.089	0.194	0.440	286.17
BERT Attn.	Yes	0.078	0.198	0.440	280.75

(b) Twitter Dataset

Table 5: Comparison of ELJST variants on IFD (a) and Twitter datasets (b)



(a) IFD Dataset

(b) Twitter Dataset

Figure 3: Performance of the different settings of ELJST model with varying T (keeping $\varepsilon = 0.3$) and ε (keeping $T = 10$ for IFD (a) and $T = 5$ for Twitter (b) datasets) for topic coherence evaluation (1,3) and topic diversity (2,4).

structure and the ability to use the semantic information through different embeddings.

Performance of ELJST under different parameter settings: In Tables 5a and 5b, we further explain the performance of different ELJST configurations on IFD and Twitter datasets. We observe similar behavior in other datasets as well. ELJST modeled with $\eta = 0$ does not use the MRF regularizer, this model shows very little improvement over the JST models. Gradual improvement is observed when we add word level or sub-word level embedding. Further, fine-tuning of embedding models on individual datasets leads drastic improvement. As shown in Figure 2, pre-trained embedding fail to capture relationship between domain specific words and their polarities. Further slight modification helps retaining the original polarity as well as understanding the connection with domain specific keywords. Of all configurations, we find the BERT attention model captures semantically the most meaningful relationships. Also it can preserve local properties (linkage between consecutive words) as well as global properties (long distance relationships), which is essential for coherent topic modelling.

In Figures 3a and 3b, we further show the performances of different ELJST configurations under different parameter settings. In ELJST we take S (number of sentiment labels) to be the same as the number of unique classes in the labelled data. Therefore, we only vary the number of topics (T) and the threshold parameter (ε). As described in Table 3, we observe that increasing ε makes the undirected graph sparse, resulting in the reduction in the the ability of MRF regularizer. We observe the downward trend in TSCS and diversity score with increasing ε. On the other hand, increasing T can lead to detected topics being more similar to each other. However, there is a trade off between coherence and diversity when we increase the number of topics.

Customer service		Rx order		Claims	
positive	negative	positive	negative	positive	negative
ELJST					
professional	dead	medicine	afford*	policy	copay
know*	information	##fill*	return	coverage	payment
customer	hang	delivery	expensive	payment	rebut
efficient	unavailable	fast	unclear	reimburs*	expensive
language	rude	free	late	authorization	denied
WS-TSWE					
customer	rude	medicine	expensive	authorization	payment
callback	difficult	doctor	return	prior	surgery
excellent	horrible	prescription	rx	network	waiting
phone	hang	clear	deliver	surgery	approval
prompt	waiting	fast	late	great	reject
RJST					
customer	rude	delivery	late	policy	network
great	drop	medicine	return	claim	cost
excellent	hang	fast	cost	hospital	expensive
timely	horrible	great	expensive	doctor	charges
conversation	pathetic	perfect	medicine	helpful	frustrating

Table 6: Top 5 words under positive and negative sentiment levels for 3 topics from IFD (* denotes wordpiece).

Qualitative Evaluation

In qualitative evaluate, we observe the top words detected by ELJST (with BERT attention weights) under different topic-sentiment pair. In Table 6 we show the topic-sentiment of our model compared to two other baselines for IFD. We show top words under positive and negative sentiment labels, where we assume rating 1 or 2 to be negative and 4 or 5 to be positive. Traditional topic models tend to pick up most frequently occurring words and word pairs under topics. Typically in e-commerce or retail domain, top frequent words are adjectives or names of products. Therefore, topics become colluded with same words, which do not show actionable insights. On the other hand, ELJST, due to the regularization factor, tend to assign high coherent word pairs under different topics. Further use of overall text sentiment helps it understand the difference between word pairs with different sentiment polarities. This leads to highly diverse set of topics for each of the sentiment classes. which lead to more coherent word pairs. As shown in Table 6, ELJST picks "knowledgeable" and "efficient" under positive sentiment for topic `Customer service`, thus it is able to understand the context as well as the correct polarity given the context. Similarly, words "return", "expensive" and "afford" are used as negative terms in the context of `Rx order` (medicine order). Both WS-TSWE and RJST use external word-sentiment lexicons, which allow them to detect "great", "excellent", "clear" under positive sentiments and similarly "expensive", "late", "horrible", "difficult" under negative sentiments. However, domain specific keywords are not often understood by these models, due to lack of knowledge in the lexicon files. On the other hand, ELJST can understand the correct sentiment polarity even for domain specific words like "knowledgeable", "unavailable", "free", "reject" etc. Additionally, with the use of fine-tuned embeddings, ELJST can put different domain-specific contextually meaningful words under relevant topic-sentiment level. With this ELJST can extract highly human interpretable results.

Conclusion

In this paper, we propose ELJST, a novel framework for joint extraction of sentiment and topics, particularly for short texts. Our proposed models are informed by the external sentiment labels which in turn, reinforce the extraction of better topics, and predict better sentiment scores. In ELJST model, we use MRF graph with word embedding representations, include attention models to compute the similarity between word in the graph. Interestingly these attention models, which have been used for the first time for this purpose, help joint topic sentiment discovery achieve the best performance. Although the use of labeled text data in the model restricts the applicability of ELJST in many applications, ELJST can be used in various applications across different industries, particularly, in the e-commerce and service based companies where sentiment/ratings are automatically labeled by the end customers. ELJST is currently deployed in a healthcare application which is helping with VoC (Voice of Customer) analysis and NPS (Net Promoter Score) improvement initiatives. In these two applications, ELJST helps in extracting granular level information from survey and complaint texts shared by customers (along with the discrete rating value on a scale of 1-5) and helps in creating value for their service and enhancing customer satisfaction.

Acknowledgements

Tanmoy Chakraborty would like to thank the support of the Ramanujan Fellowship (SERB) and the CAI, IIIT-Delhi.

References

Ali, F.; Kwak, D.; Khan, P.; El-Sappagh, S. H. A.; Ali, A.; Ullah, S.; Kim, K.; and Kwak, K. S. 2019. Transportation sentiment analysis using word embedding and ontology-based topic modeling. *Knowl.-Based Syst.* 174: 27–42. doi: 10.1016/j.knosys.2019.02.033.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR, Conference Track Proceedings*.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3: 993–1022.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguistics* 5: 135–146. URL https://transacl.org/ojs/index.php/tacl/article/view/999.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019, Volume 1*, 4171–4186. doi:10.18653/v1/n19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Dieng, A. B.; Ruiz, F. J. R.; and Blei, D. M. 2020. Topic Modeling in Embedding Spaces. *Trans. Assoc. Comput. Linguistics* 8: 439–453.

Fu, X.; Sun, X.; Wu, H.; Cui, L.; and Huang, J. Z. 2018. Weakly supervised topic sentiment joint model with word embeddings. *Knowl.-Based Syst.* 147: 43–54.

Fu, X.; Wu, H.; and Cui, L. 2016. Topic Sentiment Joint Model with Word Embeddings. In *Proceedings of DMNLP, Workshop at ECML/PKDD, 2016*, 41–48.

Goldberg, Y.; and Levy, O. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *CoRR* abs/1402.3722. URL http://arxiv.org/abs/1402.3722.

Gupta, D.; Singh, K.; Chakrabarti, S.; and Chakraborty, T. 2019. Multi-task Learning for Target-Dependent Sentiment Classification. In Yang, Q.; Zhou, Z.; Gong, Z.; Zhang, M.; and Huang, S., eds., *Advances in Knowledge Discovery and Data Mining - 23rd Pacific-Asia Conference, PAKDD'19, Macau, China, April 14-17, 2019, Proceedings, Part I*, volume 11439 of *Lecture Notes in Computer Science*, 185–197. Springer.

Hofmann, T. 1999. Probabilistic Latent Semantic Indexing. In *SIGIR '99*, 50–57. ACM.

Lee, D. D.; and Seung, H. S. 1999. Learning the parts of objects by nonnegative matrix factorization. *Nature* 401: 788–791.

Li, B.; Drozd, A.; Liu, T.; and Du, X. 2018. Subword-level Composition Functions for Learning Word Embeddings. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, 38–48. New Orleans: Association for Computational Linguistics. doi:10.18653/v1/W18-1205. URL https://www.aclweb.org/anthology/W18-1205.

Li, F.; Huang, M.; and Zhu, X. 2010. Sentiment Analysis with Global Topics and Local Dependency. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence AAAI-10*, 1371–1376.

Li, T.; Zhang, Y.; and Sindhwani, V. 2009. A Non-negative Matrix Tri-factorization Approach to Sentiment Classification with Lexical Prior Knowledge. In *Proceedings of the 47th ACL'09 and the 4th AFNLP'09*, 244–252.

Lin, C.; He, Y.; Everson, R.; and Rüger, S. M. 2012. Weakly Supervised Joint Sentiment-Topic Detection from Text. *IEEE Trans. Knowl. Data Eng.* 24(6): 1134–1145.

Mei, Q.; Shen, X.; and Zhai, C. 2007. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on KDD'07*, 490–499.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: Proceedings*, 3111–3119.

Nguyen, T. H.; and Shirai, K. 2015. Topic Modeling based Sentiment Analysis on Social Media for Stock Market Prediction. In *Proceedings of ACL*.

Poddar, L.; Hsu, W.; and Lee, M. 2017. Author-aware Aspect Topic Sentiment Model to Retrieve Supporting Opinions from Reviews. In *Proceedings of the EMNLP'17*, 472–481.

Qiang, J.; Chen, P.; Wang, T.; and Wu, X. 2017. Topic Modeling over Short Texts by Incorporating Word Embeddings. In *Advances in PAKDD, Proceedings, Part II*, 363–374.

Rahman, M. M.; and Wang, H. 2016. Hidden Topic Sentiment Model. In *Proceedings of the 25th International Conference*, WWW '16, 155–165.

Ramage, D.; Hall, D. L. W.; Nallapati, R.; and Manning, C. D. 2009. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the EMNLP'09*, 248–256.

Roy, S.; Asthana, S.; Miglani, A.; and Gupta, M. 2018. A NMF-based Approach to Topic and Sentiment Analysis of Short Texts with Prior Knowledge. Optum Tech Report, ver. 1 available at https://github.com/DSRnD/ELJST/blob/master/NMF_Topic_Sentiment_Modeling.pdf.

Sengupta, A.; Ranjan, G.; and Roy, S. 2021. LJST: A Semi-supervised Joint Sentiment-Topic1Model for Short Texts. *SN Computer Science Journal, Springer* Accepted, to appear soon, available on request.

Shi, T.; Kang, K.; Choo, J.; and Reddy, C. K. 2018. Short-Text Topic Modeling via Non-negative Matrix Factorization Enriched with Local Word-Context Correlations. In *Proceedings of the WWW18*, 1105–1114.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, 5998–6008.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document Clustering Based on Non-negative Matrix Factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on*

Research and Development in Informaion Retrieval, SIGIR '03. ACM.

Yan, X.; Guo, J.; Lan, Y.; and Cheng, X. 2013. A Biterm Topic Model for Short Texts. In *Proceedings of the 22nd International Conference*, WWW '13. ACM.