Knowledge-Based Systems 197 (2020) 105918

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Document-level multi-topic sentiment classification of Email data with BiLSTM and data augmentation

Sisi Liu, Kyungmi Lee, Ickjai Lee *

Information Technology Academy, James Cook University, PO Box 6811, Cairns, QLD 4870, Australia

ARTICLE INFO

Article history:
Received 13 December 2019
Received in revised form 11 March 2020
Accepted 13 April 2020
Available online 18 April 2020

Keywords:
Sentiment classification
Email sentiment
Multi-topic sentiment
Bidirectional LSTM
Data augmentation

ABSTRACT

Email data has unique characteristics, involving multiple topics, lengthy replies, formal language, high variance in length, high duplication, anomalies, and indirect relationships that distinguish it from other social media data. In order to better model Email documents and to capture complex sentiment structures in the content, we develop a framework for document-level multi-topic sentiment classification of Email data. Note that, a large volume of labeled Email data is rarely publicly available. We introduce an optional data augmentation process to increase the size of datasets with synthetically labeled data to reduce the probability of overfitting and underfitting during the training process. To generate segments with topic embeddings and topic weighting vectors as inputs for our proposed model, we apply both latent Dirichlet allocation topic modeling and semantic text segmentation to post-process Email documents. Empirical results obtained with multiple sets of experiments, including performance comparison against various state-of-the-art algorithms with and without data augmentation and diverse parameter settings, are analyzed to demonstrate the effectiveness of our proposed framework.

1. Introduction

As a large volume of social media text data is being transferred through daily operations, automatic quantification and extraction of useful patterns and information from these data continue to be an endless research interest. Sentiment analysis is one of the widely studied areas of text mining that analyzes intrinsic opinions and emotions from text data [1,2]. It is a prevalent way to categorize sentiment analysis tasks into document-level, sentence-level and aspect-level in terms of their granularity [1]. Among all three levels, sentiment analysis at document-level predicting the overall polarity of an opinionated document forms a basis to other granularities through comprehensive understanding of semantic structures and syntactic relations within an entire document.

Although a significant improvement has been observed in the performance of document sentiment classification with the prevalence of neural network models in recent years, several challenges still exist due to the complex semantic relations and dependency structures among words and sentences. Recent studies [3–7] are inclined to explore and capture intrinsic sentiment relations and their weighted contributions to the whole document through modeling sentences or aspects within documents to increase the classification accuracy. In particular, methods based on a hierarchical structure of documents either consider relevant positions and relational features [3,4,6], or include an attention mechanism to model aspects and their corresponding sentiments simultaneously [5,7].

Due to the unique characteristics of Email data (multi-topics, lengthy replies, formal language, high variance in length, high duplication, anomalies, and indirect relationships [8–10]), sentiment classification of Email data is different from other social media data like reviews or blogs. To better capture Email features and model sentiments in Email documents, sentiment classification for Emails at document-level with multiple topics is to be considered. We define this problem as a similar process to document-level multi-aspect sentiment classification studied in previous literature [7]. Sentiment analysis considering the concept of aspect, either as document-level multi-aspect or aspect-level sentiment classification, postulates the following features as input: a list of aspect seed terms, a fixed number of aspect ratings, or aspect labels for sentences [5,7,11]. Fig. 1 illustrates examples of multi-topic review and Email. Note that, a multi-topic Email document is observed with none of the above mentioned features. Namely, these pre-defined features are not available for Email data, instead they need to be populated from the data. Therefore, it is inappropriate to treat multi-topic Email documents as identical as aspects in reviews and other documents. This exploratory approach is more flexible, versatile and suitable than the confirmatory and pre-defined feature based approaches. Fig. 1

* Corresponding author.
E-mail addresses: Sisi.Liu1@jcu.edu.au (S. Liu), Joanne.Lee@jcu.edu.au (K. Lee), Ickjai.Lee@jcu.edu.au (I. Lee).

https://doi.org/10.1016/j.knosys.2020.105918
0950-7051/

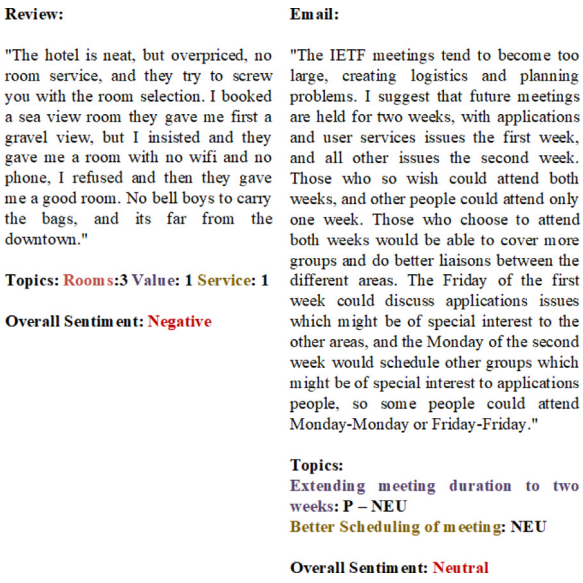


Fig. 1. Sample multi-topic review *vs.* sample multi-topic Email.

displays an example illustrating both review and Email samples where there exists a clear set of seed terms (room, value and service) for room review whilst there does not for Email data.

In consideration of the unique features of Email documents, including implicit topic-related words and no distinct difference among topics, it is hypothesized that incorporating topic features through unsupervised topic modeling is expected to improve the performance of Email document classification. In this research, we propose a Multi-Topic Bidirectional LSTM (MT-BiLSTM) model for document-level sentiment classification for Email data. The main contributions are described as follows:

- proposing a framework for document-level multi-topic sentiment classification for Email data where data augmentation is optional depends on the size of available data;
- improving semantic text segmentation techniques with Latent Dirichlet Allocation (LDA) topic modeling for converting Email into topic segments;
- developing a neural network model for multi-topic sentiment classification using Bidirectional LSTM (BiLSTM) with topic embeddings and topic weighting vectors;
- providing diverse experiments on the performance of the proposed model against various widely adopted techniques;
- evaluating the classification performance of different parameter settings for the LDA topic model, involving the input number of topics and various term weighting methods;
- examining the effectiveness and influence of the revised data augmentation technique with the proposed model.

2. Related work

As sentiment analysis aims at a deeper level of insights into the data, there still remains a gap between developed techniques and satisfactory performance despite of decades of research. In this section, we review literature in terms of Email sentiment analysis, document-level sentiment analysis with deep learning, and topic modeling in sentiment analysis that forms the conceptual and technical fundamentals of this research.

2.1. Email sentiment analysis

Literature on Email sentiment analysis is limited due to the restrictive access to labeled Email data for analysis, and the complexity of Email structure and language. Yet, research suggests that sentiment analysis for Email data provides practical and useful insights into actions and thoughts through daily operations and communications to support formal business decisions [8,12–14].

Earlier studies on Email sentiment analysis dated back a decade ago mainly focused on the visual analytics of Email sentiments with more practical applications, but less quantitative evaluations on the performance of classification methods [13–16]. In comparison, recent research focuses more on the improvement of algorithms for sentiment classification and its related tasks [8,12,17]. A study conducted by [8] introduced a sequence-based approach that takes documents represented by sentiment trajectories as inputs, and performs clustering analysis to discover common sentiment sequences and assigns sentiment labels based on each clustering result. Similarly, [12] implemented a supervised learning approach combined with linguistic features to explore different tones and emotions, frustration, politeness and formality specifically from Email messages, and [17] applied Bayesian classifier to analyze sentiments from Emails as supportive information for spam filtering.

However, these studies either lack sufficient quantitative evaluations of the classification performance, or are not directly related to Email document sentiment classification. With the advancement of machine learning algorithms, the performance of Email sentiment classification is expected to be enhanced through modeling intrinsic sentiments and semantic structures within Emails through the use of neural network models with relevant linguistic features.

2.2. Document-level sentiment analysis with deep learning

Sentiment analysis at document-level regains its attention amongst researchers with the prevalence and advancement of deep learning techniques [18,19]. Through implementing adequate neural network models, documents are modeled with deeper structures than using conventional feature-based methods. Unlike sentence-level or aspect-level sentiment analysis where more concentrated dependency between sentiment and textual information is observed, document-level sentiment analysis requires more insights into the complex intrinsic structure among sentiments and dependent words or phases. As illustrated in [6], determining weighted contribution of various parts in a document and capturing the corresponding sentiment information are key factors to improve the overall performance of document-level sentiment analysis.

Hence, techniques to model documents based on the underlying dependency structure, or to apply sentence weighting on training models are developed to capture intrinsic and semantic relations among sentences within documents. For instance, [3] presented a Rhetorical Structure Theory (RST) based neural network to improve lexicon-based sentiment analysis, and [4] proposed a Hierarchical Long Short-Term Memory (H-LSTM) model with user and product attention to incorporate user preference and product characteristics for document-level sentiment analysis.

However, as described in the previous section, due to the unique characteristics contained in Email data, existing deep neural network models, without special considerations, are not directly applicable to the document-level multi-topic sentiment classification for Email data.

2.3. Topic modeling in sentiment analysis

It becomes ubiquitous to model topics through an unsupervised aspect extraction as the research focus gradually inclined to aspect-level sentiment analysis. Literature advises that topic

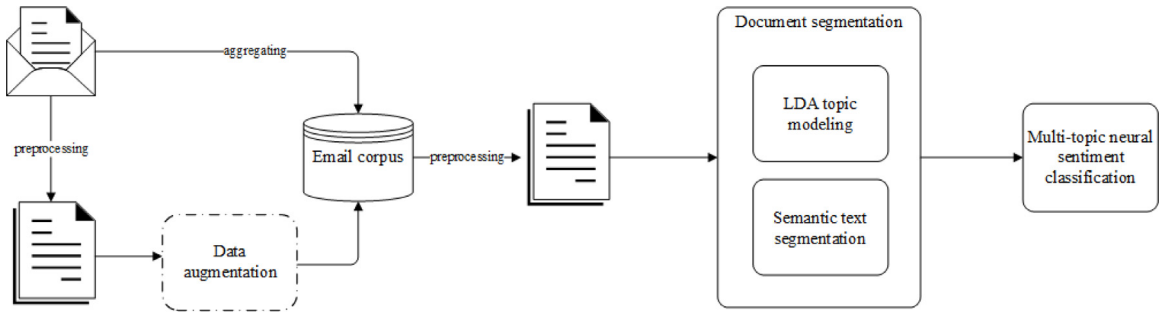


Fig. 2. Overall framework for the proposed document-level multi-topic Email sentiment analysis.

modeling methods for aspect-level or aspect involved sentiment analysis are mainly categorized into unsupervised learning-based and deep learning-based approaches [5,7,11,20].

As deep learning-based topic modeling approaches require topic labels for training, they are not suitable for this Email sentiment classification study where pre-labeled training data is unavailable. Thus, unsupervised learning-based approaches are reviewed with in-depth analysis. Among existing unsupervised topic models, LDA [21], a generative probabilistic method to model collections of discrete data such as a text corpus, is the most widely-adopted and well-developed model for sentiment analysis tasks [11,20]. For instance, [20] proposed a weakly-supervised approach that utilizes only minimal prior knowledge – in the form of seed words – to enforce a direct correspondence between topics and aspects, and [11] utilized the concept of semantic similarity to improve the effectiveness of existing LDA model in terms of aspect extraction.

As LDA operates on a full generative model and is capable of handling long-length documents, it serves as an ideal candidate for modeling topics among Email documents without pre-trained corpus and fixed lists of topic seeds.

3. Proposed framework for document-level multi-topic email sentiment analysis

In this section, we describe the details of our proposed framework for document-level multi-topic Email sentiment analysis as presented in Fig. 2. The general flow work of the proposed framework includes preprocessing of Email contents, converting documents into topic segments using LDA topic modeling and semantic text segmentation, and classifying documents into sentiment classes using the MT-BiLSTM neural network model. Note that, a data augmentation phase using a random word replacement is implemented to minimize the influence of imbalanced class distribution, to tune model parameters, and to control model fitting.

Data quality is to be ensured through adequate data preprocessing methods in order to acquire high-quality and effective analytical results. In our study, different preprocessing tasks are conducted at two text mining stages. For the data augmentation stage, to minimize the influence of noisy data and to increase the processing speed, a standard text-preprocessing step is applied to the initial raw Email data to generate clean and reliable vocabularies. Python *nltk* toolkit[1] and *re* module [22] are utilized to implement preprocessing tasks including lowercase, tokenization, spell check, stop word removal and lemmatization [23]. For the document segmentation and multi-topic neural sentiment classification stage, a thorough Email cleaning and unification step is performed considering the special characteristics of Email data. First of all, we utilize a pre-developed *EmailParser*[2] package to recognize and remove salutations and signature blocks from raw Email messages. In addition, we filter out duplicated content portions from Emails that begin with or contain keywords *original*, *re* : or *reply*, and *fw* : or *forward* in either Email title or content, using the same Python regular expression operations. To maintain the syntactic relations among phases and semantic integrity of sentences in Emails, a minimal level of standard text-preprocessing, including tokenization and lemmatization, is undertaken at this stage.

3.1. Data augmentation with random word replacement

A review on deep learning indicates that in addition to data quality, data quantity also has a significant impact on the performance of neural network models [24]. Moreover, a relatively balanced class distribution of datasets is essential to ensure moderate constraint of model training and stable variance of model estimation [25]. However, due to the lack of a large volume of publicly available labeled Email datasets, it is required to consider an adequate use of machine learning techniques to automatically generate synthetic data. Inspired by past studies that utilize data augmentation methods to increase the size of dataset [26–28], we implement a hybrid method with a combination of *k*-Nearest-Neighbor (*k*NN) classifier with word embeddings and WordNet (WN) lexicon [29] to handle unique Email data.

To be more specific, we first generate a word replacement dictionary that contains words in the vocabulary and their synonyms or terms of similar usage using word embeddings with *k*NN classifier. A post-processed Email corpus is tokenized into a list of vocabularies, and each document is transferred into a collection of numeric vectors using pre-trained Glovec word embeddings [30]. We apply *k*NN classifier to the vectorized corpus to identify the first 5 nearest neighboring words for each term in the vocabulary, and store them in a dictionary. Then, the WN lexicon and its synonym thesaurus are utilized to filter out improper replacement terms, such as acronyms, generated by word embeddings, and to expand the coverage of the existing dictionary. If a key word in the dictionary is indexed in the WN lexicon, then any of its values that do not return as synonyms by WN is removed, and additional synonyms that do not exist as values are appended. An example of words in the vocabulary and their replacement terms in the dictionary is presented in Table 1.

Finally, we construct synthetic documents using the above generated dictionary. To determine the probability of a word to be replaced, a threshold value δ of 0.5 is defined. As suggested by [28], using a random synonym with a replacement rate less than 20% for each sentence yields a better performance, we test with different probabilities and set a final threshold value δ to 0.5

[1] https://www.nltk.org.

[2] https://github.com/mynameisvinn/EmailParser

Table 1
Sample words and their replacement terms.

Word in Emails	Word in Dictionary
accused	['accuse', 'impeach', 'incriminate', 'criminate', 'charge']
broadly	['loosely', 'generally']
email	['mail', 'twitter', 'facebook', 'message']
grateful	['thankful', 'thank', 'glad', 'happy', 'wish']
educational	['education', 'academic', 'learning', 'teaching', 'community']
unfortunately	['unluckily', 'regrettably', 'alas']
enroll	['inscribe', 'enter', 'enroll', 'recruit']
plot	['game', 'patch', 'diagram', 'plat']

Table 2
Example sentences and their synthetic sentences.

Original	Synthetic
I don't get any unusual code.	I don't get any strange code. I don't have any strange cipher.
Actually I think there are some potentially interesting effectual ramifications.	Actually I recall there are some potentially interesting legal ramifications. Actually I suppose there are some potentially interesting sound ramifications.
It's a good piece of work.	It's an effective part of work. It's a good nibble of work.

that results in a coverage of replacement rate for each document within a range of 5% to 20%. Apart from stop words, each word in a document is assigned with a random value that is compared with δ. If a word has a random number greater than δ and exists as a key word in the dictionary, then it is replaced by one of its random values. Table 2 lists some example sentences with their synthetic ones.

3.2. Document transformation into topic segments

The main component of our proposed framework is the document transformation phase that aims at modeling documents based on topic representations and splitting documents into topic segments. In brief, this phase is further divided into a LDA topic modeling process and a semantic text segmentation process. For each Email document, the former step returns a list of topics with sets of keywords as representations, and then the number of topics is treated as an input parameter for the text segmentation method in the latter step to split the document into n number of segments.

3.2.1. LDA topic modeling

As discussed in previous sections, LDA, a generative topic model designed on Bayesian probabilistic theory, is a widely adopted technique for modeling text corpora with topic probabilities [21]. Gensim [31], a well-developed Python library for various statistical modeling, is utilized to implement various functions involved in the LDA topic modeling process. To generate topic representations for a collection of N documents in a corpus, a *LDAModel* object is to be initialized with documents vectorized using *TFIDFVectorizer*[3] and a value α to specify the number of topics as input parameters. Once the LDA model is constructed, the *get_document_topics* function is utilized to return the topic representation with a list of topics and their probabilistic distributions for each document. A minimum probability threshold value θ is set on the basis of the adjusted mean calculated by the summation of mean and skewness for asymmetric unimodal distribution. The relevant mathematical formulas are presented below:

$$\begin{aligned} \bar{\mathcal{P}} &= \frac{1}{n*m}\sum_{i}^{n}\sum_{j}^{m}\mathcal{P}_{ij} \\ \sigma &= \sqrt{\frac{1}{n*m-1}\sum_{i}^{n}\sum_{j}^{m}(\mathcal{P}_{ij}-\bar{\mathcal{P}})^2} \\ \theta &= \bar{\mathcal{P}} + \sum_{i}^{n}\sum_{j}^{m}\left(\frac{\mathcal{P}_{ij}-\bar{\mathcal{P}}}{\sigma}\right)^3, \end{aligned} \tag{1}$$

where $\mathcal{P}_{ij}$ represents the probability of the jth topic that belongs to the ith document for $1 \leq j \leq \alpha$ and $1 \leq i \leq N$. A revised list of topic representations for each document is generated by removing topics with probability less than θ, and stored with the number of topics assigned to each document as part of computation in the next step.

3.2.2. Semantic text segmentation

In this step, a pre-developed package *TextSegment*[4] is first utilized to perform text segmentation with the number of topics as input parameters. In general, the segmentation process is performed by the *get_segment_texts* function. The basic working mechanism operates on the greedy heuristic algorithm that chooses the best split point iteratively through computing weighted distances of words to a segment centroid. The weighted distance of a word is computed by multiplying an entropy with a cosine distance between average centroid and word embeddings using pre-trained Glovec [30] (to be consistent with the hereinafter neural model). Eq. (2) indicates the arithmetic calculation for individual entropy ($ent(i)$) and weighted distance($wd(i)$).

$$\begin{aligned} entropy(i) &= -\frac{f_i}{f} * \log_2\frac{f_i}{f} \\ wd(i) &= entropy(i) * cos\left(\frac{e_i * entropy(i)}{i+1}, e_i\right), \end{aligned} \tag{2}$$

where f_i and f represent the frequency of ith word and the sum of frequencies for all words in a document, respectively. e_i represents the word embeddings of ith word in a document, and

[3] Experiments were conducted with three document vectorization methods: n-gram, Word2vec and TF–IDF, among which TF–IDF yields the best results.

[4] https://github.com/ReemHal/Semantic-Text-Segmentation-with-Embeddings.

$cos()$ function computes the cosine distance between $\frac{e_i * entropy(i)}{i+1}$ (centroid) and e_i.

Subsequently, we apply the cosine similarity measurement to TF–IDF vectorized words in topic segments and topic representations to assign each topic segment to the corresponding topic in each component. To explain the process in details, a Pseudocode 1 of transferring Email documents into topic segments (*EmailTTS*) is presented. Denote $\mathcal{ED}$ as a collection of Email documents composed of messages $\{ed_1, ed_2, \ldots, ed_n\}$, and for each Email document $ed_i \in \mathcal{ED}$, denote $\mathcal{TD}$ as a list of topics $\{td_1, td_2, \ldots, td_m\}$ assigned, and $\mathcal{KW}$ as a list of keywords $\{kw_1, kw_2, \ldots, kw_p\}$ that represents each topic $td_j \in \mathcal{TD}$; $\mathcal{TS}$ as a list of topic segments $\{ts_1, ts_2, \ldots, ts_v\}$ generated, and $\mathcal{TW}$ as a set of token words $\{tw_1, tw_2, \ldots, tw_q\}$ that belongs to each topic segment $ts_j \in \mathcal{TS}$.

Pseudocode 1 EmailTTS

Input: A set of post-processed Email documents $\mathcal{ED}$;
Output: Each Email document $ed_i \in \mathcal{ED}$ represented by a list of topics $\mathcal{TD}$, in which a topic is associated with a keyword list $\mathcal{KW}$ and a token list $\mathcal{TW}$;
for each Email document $ed_i \in \mathcal{ED}$ **do**
  Tokenize ed_i into a collection of words;
  Apply *TFIDFVectorizer* to ed_i;
  Store vectorized document as Email corpus $\mathcal{C}$;
end for
Initialize a *LDAModel* object;
Train on the Email corpus $\mathcal{C}$;
Initialize two empty dictionaries $\mathcal{TD}$ and $\mathcal{N}$;
for each Email document $ed_i \in \mathcal{ED}$ **do**
  Apply *get_document_topics*() to ed_i;
  Return a temporary topic list $\mathcal{TD}$;
  for each topic $td_j \in \mathcal{TD}$ **do**;
    Apply *TFIDFVectorizer* to the corresponding keyword list $\mathcal{KW}$;
    Compute average TF–IDF value α for $\mathcal{KW}$;
    Get a probability p_j for the topic;
    if $p_j < \theta$ **then** /*θ is defined in Equation (1)*/
      Remove topic td_j from $\mathcal{TD}$;
    end if
  end for
  Append the number of topics n_i to $\mathcal{N}$;
  Initialize a *TextSegment* object;
  Apply *get_segment_texts*() to ed_i with n_i as input;
  Return a topic segment list $\mathcal{TS}$;
  for each topic segment $ts_j \in \mathcal{TS}$ **do**;
    Apply *TFIDFVectorizer* to the corresponding token list $\mathcal{TW}$;
    Compute average TF–IDF value β for $\mathcal{TW}$;
    Compute cosine similarity between α and β; e
    Assign $\mathcal{TW}$ to td_j;
  end for
end for

3.3. Multi-topic neural sentiment classification

The proposed multi-topic neural sentiment classification model is built upon a topical structure with two BiLSTM layers introduced by [32] as the basis. Fig. 3 illustrates the overall structure of the proposed MT-BiLSTM. The outputs of the first topic-level BiLSTM layer are concatenated with a topic embedding layer and fed into a document-level BiLSTM that is multiplied by a topic weighting vector (a weighted representation of topic segments with a given topic).

3.3.1. Document and topic representation

Word embedding is a technique that maps terms into numeric vectors to precisely capture semantic information and contextual similarity for text mining tasks [33]. To obtain document and topic representations for input, we pad all topic segments to length l using padding tokens, and insert dummy topic segments and topics to ensure documents are represented by a fixed number of topic segments and topics.

Given a set of input $\mathcal{ED}$ containing n documents, each document is represented by a vectorized three-dimensional matrix denoted by $ed_i \in \mathbb{R}^{d \times l \times t}$ where d refers to the embedding dimensions of words, l refers to the maximum length of a topic segment, and t refers to the maximum number of topics in the corpus.

Each topic segment is associated with a topic represented by a fixed length of keywords p and a topic weighting vector w with length 1. Topic embeddings for each topic is calculated by averaging a dimension of d_t of word embeddings for all keywords $[\frac{\sum_{w=1}^{p} e_{w1}}{p} : \frac{\sum_{w=1}^{p} e_{wd_t}}{p}]$. Hence, given two sets of input $\mathcal{TL}$ and $\mathcal{W}$ containing n topic and weighting lists, respectively, each topic is represented by a vectorized two-dimensional matrix denoted by $tl_i \in \mathbb{R}^{d_t \times t}$ and each weighting is represented by a vectorized matrix denoted by $w_i \in \mathbb{R}^{t}$.

3.3.2. Bidirectional LSTM

Long Short-Term Memory (LSTM) network [34] is an extended variant of traditional feed-forward neural network. The most apparent advantage of LSTM over other recurrent neural models is its ability of handling vanishing and exploding gradients problem. LSTM manages to capture long-term dependencies from sequential structured data through iteratively updating the memory state from a series of building blocks. Each building block centers a memory cell state that is updated by recurrent input information filtered by three functional gates using sigmoid activation function. A forget gate manipulates the update of the current memory state through either forgetting or memorizing recurrent inputs, and an input gate and an output gate controls the flow of recurrent inputs through either erasing or keeping the current cell state [34].

BiLSTM [32] is developed based on two LSTM layers. It computes not only the hidden state of a forward sequence, but the hidden state of a backward sequence as well. With two LSTM layers that process data in both directions, BiLSTM is capable of modeling sequential dependencies of a piece text from both previous and successive context. Denote $\overrightarrow{\mathcal{H}}$ as a series of hidden states $[h_1, h2, \ldots, h_t]$ generated by a forward sequence, and $\overleftarrow{\mathcal{H}}$ as a series of hidden states $[h_t, h_{t-1}, \ldots, h_1]$ generated by a backward sequence. A BiLSTM computes the output sequence V_t at a given time step t by concatenating a $\overrightarrow{h_T} \in \overrightarrow{\mathcal{H}}$ and a $\overleftarrow{h_T} \in \overleftarrow{\mathcal{H}}$, which is mathematically denoted as:

$$\begin{aligned} V_t &= \overrightarrow{h_T} \oplus \overleftarrow{h_T} \\ &= W_{\overrightarrow{h_v}} \cdot \overrightarrow{h_T} + W_{\overleftarrow{h_v}} \cdot \overleftarrow{h_T} + b_V, \end{aligned} \tag{3}$$

where W refers to a weight matrix, and b refers to a bias vector for the corresponding input hidden vector.

3.3.3. Document-level multi-topic Bi-LSTM

In our proposed model, we first apply a topic-level BiLSTM to each topic segment represented by word embeddings. Results are two sequences of hidden vectors denoted by two matrices $\overrightarrow{\mathcal{H}_{ts}} \in \mathbb{R}^{h \times l}$ and $\overleftarrow{\mathcal{H}_{ts}} \in \mathbb{R}^{h \times l}$ where h is the size of hidden layers and l is the length of the given topic segment. Then, we concatenate a topic embedding matrix $\mathcal{E}_t \in \mathbb{R}^{d_t}$ to the final hidden state

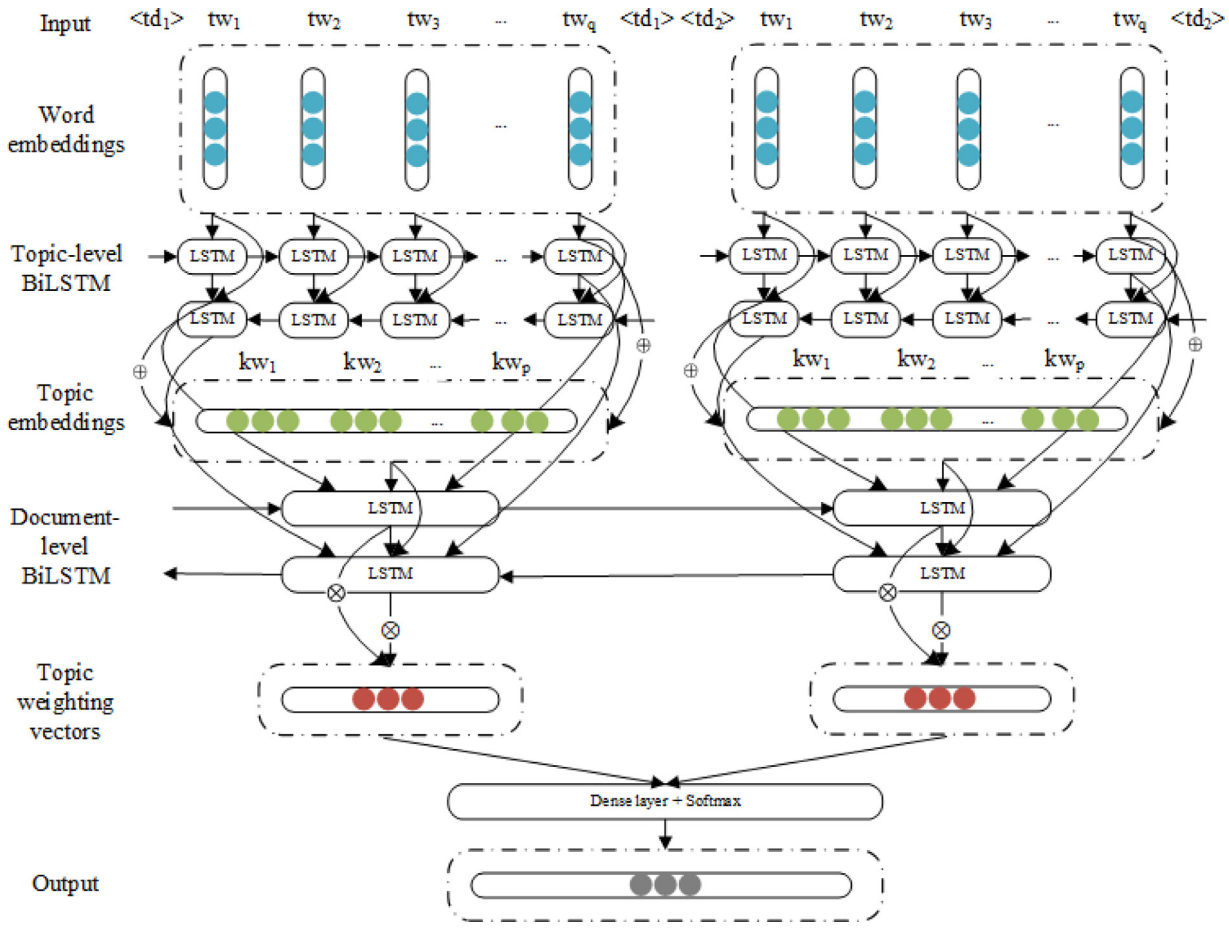


Fig. 3. Overall model structure of MT-BiLSTM for document-level sentiment analysis. Given a sample document ed_i that has two topics $\langle td_1 \rangle$ and $\langle td_2 \rangle$. A topic-level BiLSTM is applied to each topic segment that is represented by word vectors $tw_1, tw_2, tw_3, \ldots, tw_q$ with length q. A time-distributed representation of topic segments is concatenated with a topic embedding layer that is represented by keyword vectors $kw_1, kw_2, \ldots, kw_q$ using $\oplus$ operator, and fed into a document-level BiLSTM. A probability distributed topic segment is further multiplied by a topic weighting vector using $\otimes$ operator, and fed into a final dense layer for output.

of both forward sequence $\overrightarrow{h_{ts}}$ and backward sequence $\overleftarrow{h_{ts}}$. The output vector $\mathcal{H}_t$ is given by:

$$\mathcal{H}_t = [\overrightarrow{h_{ts}} \oplus \mathcal{E}_t, \overleftarrow{h_{ts}} \oplus \mathcal{E}_t], \tag{4}$$

where $\mathcal{H}_t \in \mathbb{R}^{(h+d_t)\times 2}$ is a vector representation of each topic segment concatenated with topic embeddings in a document.

Subsequently, we apply a document-level BiLSTM to each document represented by topic segment vectors, resulting in two sequences of hidden vectors denoted by matrices $\overrightarrow{\mathcal{H}_T} \in \mathbb{R}^{(h+d_t)\times 2\times t}$ and $\overleftarrow{\mathcal{H}_T} \in \mathbb{R}^{(h+d_t)\times 2\times t}$. Finally, a *softmax* layer is implemented to output the probability distribution of each weighted topic segment scaled by a topic weighting vector w_i to the overall sentiment:

$$\begin{aligned} &\mathcal{H}_d = [\overrightarrow{\mathcal{H}_T} \otimes w_i, \overleftarrow{\mathcal{H}_T} \otimes w_i], i \in (1, t), \\ &y = softmax(W_d * \mathcal{H}_d + b_d), \end{aligned} \tag{5}$$

where $\mathcal{H}_d \in \mathbb{R}^{h+d_t}$, and $\otimes$ reflects a point-wise multiplication operator that multiplies the topic weighting value w_i with each element in the matrix $\overrightarrow{\mathcal{H}_T}$ and $\overleftarrow{\mathcal{H}_T}$. W and b refer to a weight matrix and a bias vector for the *softmax* function, respectively.

4. Empirical experiments

In this section, we elaborate on the preparation and adjustment of datasets and parameter settings used for different techniques under study. As Email sentiment classification is rarely studied, experimental results are reported on comparisons of the proposed neural classification model with various widely adopted techniques at different levels: involving lexicon-based, machine learning, and deep learning approaches. Additionally, we justify the options for term weighting techniques and parameters involved in the LDA topic modeling through a comparative analysis on the classification performance with our proposed model.

4.1. Datasets

As large size of sentiment labeled Email datasets are rarely publicly available, we perform alpha experiments on three medium-sized Email datasets, among which two are public datasets and one is a private archive. Initially, three Email datasets are labeled based on different standards and number of classes. We adjust the class labels through statistical modeling methods for the two public datasets to ensure the reliability and authentication of the classification performance. A brief justification on the sources of three Email datasets, and the corresponding label conversion process, is described as:

BC3 dataset. Abbreviated as BC3, the first dataset is extracted from the British Columbia Conversation Corpora [35] that includes 255 messages derived from 40 email threads and labeled with four sentiment classes, involving *positive* (P), *negative* (N), *both* (PN), or *neither* (X), on a sentence basis. To acquire a document level three-class label, a label conversion is performed with transforming a *both* (PN) label into one *positive* (P) and one *negative* (N) label, and replacing a *neither* (X) label with a

Table 3
Summary of distributions over three labels of three datasets (NoE: Number of Emails).

Dataset	NoE	Labeled NoE	MaxTS	MaxNT
BC3	255	Positive — 147 Negative — 29 Neutral — 79	214	4
EnronFFP	960	Strongly Negative — 172 Negative — 214 Neutral — 574	105	3
PA	600	Positive — 150 Negative — 128 Neutral — 322	98	3

neutral (*N*) label. Then, the majority voting, a well-adopted discriminative modeling technique for text mining [36], is utilized. The dataset ends up with a class distribution of 147 positive, 29 negative, and 79 neutral Emails, respectively.

Enronffp dataset. Abbreviated as EnronFFP, the second dataset a subset of the well-known Enron Email corpus that contains 960 messages labeled by [12]. As the study focuses on quantifying both feelings and tones from Email messages, it preforms three sets of labeling in terms of frustration, formality and politeness. Suggested by psychological studies that frustration is a measurement for negative feeling on the basis of seven visual analog scales [37,38], it is justifiable to use frustration labels as a reference to identify negative Emails from the dataset. To obtain a three-class label that is consistent with other datasets, we convert the labels into *Strongly Negative* (*SN*), *Negative* (*N*), and *Neutral* (*Neu*) based on a threshold value μ of -0.7 that sets the boundary between *SN* and *N*. The value is calculated with an average frustration value (within an interval of $[-2, -1, 0]$) of summation over 10 annotators for all Emails. The dataset results in a final class distribution of 172 strongly negative, 214 negative, and 574 neutral Emails, respectively.

PA Dataset. Abbreviated as PA, the third dataset contains 600 messages originated from a private Email account that belongs to one of the authors. Each Email is manually labeled by a set of three annotators, including the sender and the recipient of this Email, and an independent third party. Each annotator is given options of either assigning an Email with sentiment orientation to a value in the range of $[-1, 1]$ or an Email without sentiment to a value of 0. The final label of each Email is determined by the weighted average value among all three annotators where 50% for the sender, 30% for the recipient, and 20% for the independent third party. A *Positive* (*P*) label is attached if the weighted average value is above 1, a *Negative* (*N*) label if below 0, and a *Neutral* (*Neu*) label if equal to 0. The dataset exhibits a class distribution of 150 positive, 128 negative, and 322 neutral Emails, respectively.

Table 3 summarizes the class distribution and length features of the three datasets. Noted that the maximum length of a topic segments *MaxTS*, and the maximum number of topics *MaxNT* vary with different input parameters for the LDA model. Figures reported here are generated based on 10 topics with *TF–IDF* term weighting method for the topic model where a truncated *MaxTS* performs better than the original maximum length of topic segments.

4.2. Experimental settings

We conducted effectiveness evaluations of proposed MT-BiLSTM model and its variants against recent approaches for Email sentiment classification [8,12,17], lexical and machine learning based baseline approaches, and state-of-the-art neural network based approaches for document sentiment classification. A brief description of each method is presented as follows:

- **Baseline**: a lexical-based approach that predicts sentiment polarities by computing BoWs weighted SWN lexicon features;
- **TRACLUS** [8]: a sequence-based approach that performs clustering on documents transformed into sentiment trajectories using a revised TRACLUS algorithm developed by [8];
- **SVM** [39]: a benchmarked supervised-learning approach that yields the best performance in comparison to others. Aggregated word embeddings proposed by [40] are used as features;
- **MLP** [41]: a classic feed-forward neural network model with three layers of perceptrons controlled by a nonlinear activation function;
- **LSTM** [34]: an extended variant of traditional feed-forward neural network model that operates on a series of building blocks that contains a memory cell state and three multiplicative gates. A hidden state of 100 is set as the input parameter;
- **CNN** [42]: a classic variant of conventional deep neural networks that implements convolutional filters with learned weights and bias. A window size of [3, 4, 5] with a filter size of 32 for each convolutional layer is defined as input parameters;
- **BiLSTM** [32]: a bidirectional LSTM developed by [32] with a concatenated layer of one forward LSTM and one backward LSTM and a hidden state of 100;
- **H-BiLSTM** [5]: a hierarchical-based bidirectional LSTM that is composed of a sentence-level BiLSTM layer and a document-level BiLSTM layer. Both layers are set to a hidden state of 100;
- **HAN** [6]: a hierarchical attention network developed based on a hierarchical structure with bidirectional Gated Recurrent Unit (GRU) and attention mechanism at both word and sentence level. A hidden state of 100 is set for both bidirectional GRU layers and attention layers.

Table 4
Hyper-parameter settings for the proposed MT-BiLSTM model of three datasets.

	BC3	EnronFFP	PA
Hidden state	100	150	100
Dropout probability	0.3	0.5	0.3
Learning rate	0.01	0.01	0.01
Batch size	32	64	32
No. of epochs	15	50	30

Experimental results with two sets of parameters involved are reported in the following section. Table 4 summarizes the hyper-parameter settings of the neural network models for each Email dataset. Note that, the same *batch size* and *num epochs* of each dataset are used for all deep learning based models. Additionally, we use the pre-trained GloVec [30] with a dimension of 100 for both word embeddings and topic embeddings considering its adequate coverage and moderate processing time.

Experiments are undertaken with the 10 fold cross-validation in a view of the medium size of the datasets. Evaluation criteria involve accuracy and Root Mean Squared Error (RMSE) that are averaged from 10 sets of experiments as standard matrices for multi-class classification tasks. Effectiveness of the proposed model is justified through higher accuracy and lower RMSE than

Table 5
Overall performance comparison of various methods under study. The symbol ∗ indicates the best result from various experimental settings. Bold texts indicate results to be highlighted.

Classifier	Dataset					
	BC3		EnronFFP		PA	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
Baseline	0.373	1.262	0.205	1.216	0.308	1.016
TRACLUS [8]	0.592	0.876	0.579	0.714	0.793	0.397
SVM [39]	0.584	0.882	0.595	0.675	0.623	0.706
MLP [41]	0.789	0.506	0.582	0.651	0.649	0.607
LSTM [34]	0.852	0.461	0.586	0.652	0.588	0.642
CNN [42]	0.852	0.461	0.598	0.634	0.653	0.606
BiLSTM [32]	0.873	0.512	0.742	0.552	0.788	0.442
H-BiLSTM [5]	0.874	0.512	0.739	0.574	0.817	0.396
HAN [6]	0.861	0.556	0.721	0.621	0.742	0.508
Topic-BiLSTM ∗	0.903	0.317	0.770	0.472	0.841	0.377
Topic-TE-BiLSTM ∗	0.913	**0.282**	0.781	0.459	0.852	0.359
Topic-TW-LSTM ∗	0.897	0.319	0.779	0.470	0.850	0.372
MT-BiLSTM ∗	**0.918**	0.295	**0.788**	**0.439**	**0.859**	**0.355**

other comparative approaches. Formulas are given below where n is the number of objects:

$$Accuracy = \frac{\sum_i^n (predict_i - true_i)}{n}, \quad RMSE = \sqrt{\frac{\sum_i^n (predict_i - true_i)^2}{n}}. \tag{6}$$

4.3. Classification results

The foremost group of classification performance is presented and profiled on the basis of our proposed model with three base variations: including *Topic − BiLSTM* for MT-BiLSTM model without topic embeddings and topic weighting vectors, *Topic − TE − BiLSTM* for topic embeddings incorporated MT-BiLSTM models, and *Topic − TW − BiLSTM* for topic weighting incorporated MT-BiLSTM models, compared with other algorithms described in Section 4.2. Table 5 concludes the performance of different algorithms for three datasets respectively where the outcomes of our proposed model and its three variants are opted from the best among different parameter settings of the LDA topic model.

Major findings of empirical results in Table 5 can be summarized into the following three aspects:

1. First, our proposed *MT − BiLSTM* model obtains the highest accuracy percentage of 91.8%, 78.8% and 85.9% for all three datasets, respectively. Though the lowest RMSE value of 0.282 for BC3 dataset is acquired by *Topic − TE − BiLSTM* model, *MT − BiLSTM* model manages to achieve the lowest RMSE value of 0.439 and 0.355 for the rest two datasets. These empirical results demonstrate the effectiveness of our proposed *MT − BiLSTM* model in terms of Email document sentiment classification.
2. Second, the hypothesis of classifying Email sentiments through document-level multi-topic point of view is justified. Our proposed model achieves significant improvements in the performance over *H − BiLSTM* that obtains relatively the best results among all listed state-of-the-art methods with an increase of 4.4%, 4.9% and 4.2% in accuracy for each dataset among the three.
3. Last, results indicate that topic-based neural network models incorporating topic related features accurately predict sentiments at document-level, and procure better classification performance than other document-based algorithms. For instance, the base variation of the proposed model Topic-BiLSTM acquires an accuracy rate of 90.3%, 77.0% and 84.1%, and a RMSE rate of 0.317, 0.472 and 0.377 for BC3, Enron FFP and PA, respectively, which outperforms all baseline methods, such as *SVM* (p-value = 0.013 for accuracy and p-value = 0.038 for RMSE with 95% confidence), *CNN* (p-value = 0.043 for accuracy and p-value = 0.01 for RMSE with 95% confidence) and *H − BiLSTM* (p-value = 0.002 for accuracy with 99% confidence and p-value = 0.087 for RMSE with 90% confidence).

Two additional groups of experiments are conducted on the effect of our revised data augmentation technique and that of LDA topic modeling with different parameter settings for further evaluations on the overall framework proposed in this study.

The proposed method is composed of preprocessing phase, document segmentation phase and neural classification phase where the document segmentation phase further contains LDA topic modeling phase and semantic text segmentation phase. Their Bio-O time complexities are: $O(n)$, $O(nmt)$, $O(sr + skr)$, $O((s + k + 1)^r)$, respectively, where n represents the number of Email documents, m represents the number of words in Email documents, t represents the number of initial topics, s represents the number of topic segments, r represents the number of filtered topics and k represents the number of topic keywords. The space complexity for neural sentiment classification phase is $O(s^h dr + k^d r)$, where h represents the number of hidden states and d represents the dimension of word embeddings.

4.3.1. Effect of email data augmentation

As an optional phase in our proposed framework, the purpose of implementing data augmentation is to handle imbalanced class distribution and to tune neural model parameters with sufficient data. Therefore, two sets of augmented data are synthesized with one set maintaining the same class distribution and the other set reformulated for a balanced distribution with an undersampling technique. Each set contains a series of data augmented based on a ratio within the interval of [10, 20, 30, 50, 100] compared to the original.

Representative classification results with data augmentation are presented in Table 6 and Fig. 4. As demonstrated in Table 6, topic-based neural network models achieve better performance with augmented datasets than the original. For example, *MT − BiLSTM* results in an accuracy rate of 93.5%, 88.8% and 87.4%, which is equivalent to an increase rate of 1.7%, 10.0% and 1.5%, for each individual in all three datasets. Fig. 4 illustrates the classification accuracy of MT-BiLSTM model with augmented data, both balanced and imbalanced, at different ratios where dot lines indicate the benchmark values of original datasets. In accordance with the empirical results, augmented data has remarkably positive influence on the performance of neural network models, and balanced augmented data even outperforms the imbalanced one.

In terms of statistical evaluations, t-test results reflect a significant increase of accuracy rates on data augmentation techniques for BC3 (p-value = 0.004) and PA (p-value = 0.008) with 99% confidence, and for Enron FFP (p-value = 0.016) with 95% confidence.

4.3.2. Effect of LDA topic modeling with different parameter settings

As the proposed MT-BiLSTM model is notably dependent on the topic-level inputs generated by the LDA topic modeling, an in-depth analysis on comparative experiments of two parameters

Table 6
Performance comparison of topic-based neural network models with original and augmented datasets. Results for augmented datasets are achieved using a ratio of 100 : 1 to its original.

Model	Dataset											
	BC3				Enron FFP				PA			
	Original		Augmented		Original		Augmented		Original		Augmented	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
Topic-BiLSTM	0.903	**0.317**	**0.934**	0.347	0.770	0.472	**0.797**	**0.468**	0.841	0.377	**0.859**	**0.375**
Topic-TE-BiLSTM	0.913	0.282	**0.935**	**0.270**	0.781	0.459	**0.859**	**0.405**	0.852	**0.369**	**0.858**	0.377
Topic-TW-BiLSTM	0.897	0.319	**0.931**	**0.291**	0.779	**0.470**	**0.824**	0.522	0.850	0.372	**0.870**	**0.361**
MT-BiLSTM	0.918	0.295	**0.935**	**0.259**	0.788	0.439	**0.888**	**0.434**	0.859	0.355	**0.874**	**0.354**

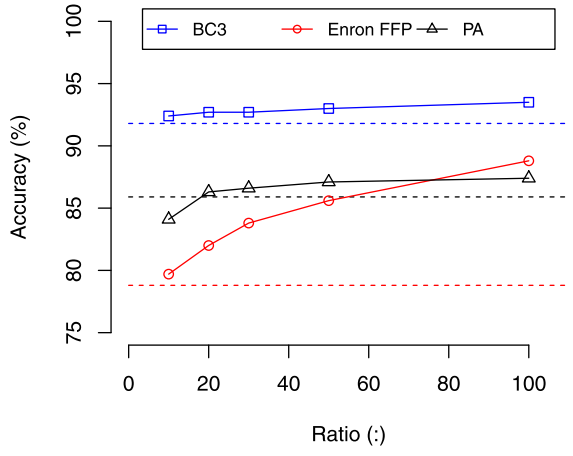


Fig. 4. Classification accuracy with regard to different levels of augmentation: a) the number of Emails augmented based on ratio; b) the number of Emails augmented based on balanced ratio.

that influence the outputs of LDA model is undertaken. We first illustrate comparative results of different term weighting methods that mainly influence the input topic weighting vectors for MT-BiLSTM model. Since different term weighting methods generate varied features as inputs for LDA model, different topic weighting vectors and topic distributions are obtained accordingly. Table 7

Table 7
Classification performance with regard to different term weighting methods.

Method	BC3		Enron FFP		PA	
	Accuracy	RMSE	Accuracy	RMSE	Accuracy	RMSE
TF–IDF	**0.918**	**0.295**	**0.788**	**0.439**	**0.859**	**0.355**
n-gram	0.837	0.476	0.729	0.574	0.790	0.497
w2v	0.891	0.325	0.777	0.450	0.852	0.420

summarizes the classification performance of MT-BiLSTM model with different term weighting methods, including $TF–IDF$, $n-gram$, $w2v$, in which $TF–IDF$ is the final option for our proposed model as it yields the best results.

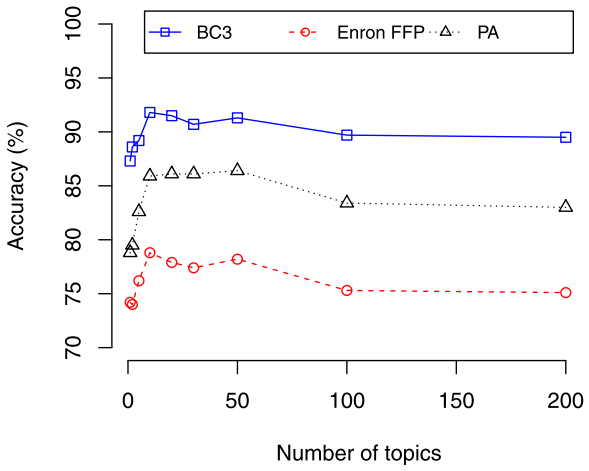
We then evaluate the influence of LDA with different number of topics as an input parameter for all three datasets with a fixed number of keywords of 10 for topic embeddings. Fig. 5 displays the performance comparison of LDA model with an input number of topics within the interval of $[1, 2, 5, 10, 20, 30, 50, 100, 200]$ where the results of an input topic number of 1 are equivalent to those of *BiLSTM* model.

As shown in Fig. 5, LDA with an input topic number of 10 achieves the highest accuracy rate of 91.8% and 78.8%, and the lowest RMSE rate of 0.295 and 0.439 for BC3 and EnronFFP dataset. For PA dataset, the highest accuracy rate of 86.1% is achieved with a topic number of 20. Judging by the effectiveness, we ultimately choose a topic number of 10 for reporting the overall classification results of all three datasets.

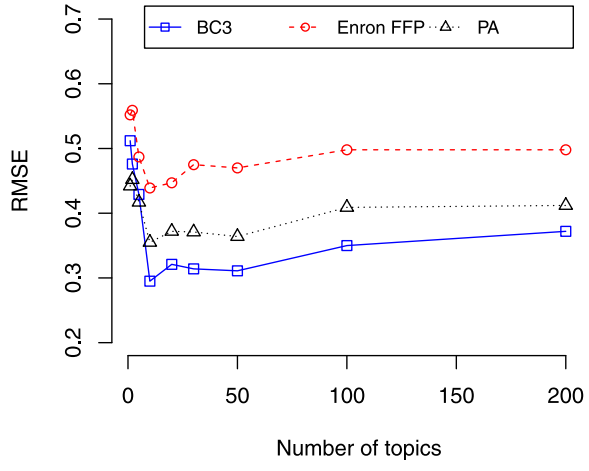
5. Conclusion and future work

We propose a document-level multi-topic sentiment analysis framework for Emails where an optional data augmentation phase is utilized due to insufficient data for training neural network models. We propose MT-BiLSTM to model structural dependencies on a topic-level within documents. We use the LDA topic modeling with a semantic text segmentation to transfer documents into topic segments where each topic segment is associated with a topic representation with a probability distribution. Along with documents represented by topic segments, topic embeddings and topic weighting vectors obtained during the LDA modeling process are utilized as additional inputs for the proposed model. A topic-level BiLSTM concatenated with topic embeddings is applied to generate a vector representation of topic segments, and a document-level BiLSTM scaled by a topic weighting vector is applied to generate a weighted probability distribution of each topic segment for output.

Empirical experiments demonstrating high classification accuracies and low error rates prove the effectiveness of our proposed model over existing approaches. Our results reflect an advantage of topic-based models over conventional document-based models. Subsequently, overall better classification performance of



(a)



(b)

Fig. 5. Classification performance with regard to the number of topics: (a) accuracy with regard to the number of topics; (b) RMSE with regard to the number of topics.

MT-BiLSTM model with augmented data than original data indicates a positive influence of training neural network models with a large size of data and balanced class distribution. In addition, we conduct further evaluations on the effect of parameter settings for the LDA topic modeling that quantitatively justifies the options used in the model.

One of the influential factors to the classification performance of our proposed algorithm is the determination of the number of topic segments for Email data. Since this number is closely related to the outputs of LDA topic modeling, and our experimental results also reflect that the proposed algorithm obtains the most preferable performance on three Email datasets with different initial numbers of topics. It is inferred that the proposed method is dependent on the input parameter of LDA topic modeling. Hence, one possible improvement of the proposed method is to incorporate automatic search algorithm for appropriate input parameter for LDA topic modeling. As LDA topic modeling operates on the Gibbs sampler that estimates the posterior probability of topic distribution through iterating sampling topic assignments of training document, to adjust LDA topic modeling phase in the proposed method to handle Email documents with unseen topics is another task for potential improvement.

Although we introduce data augmentation mainly for the purposes of minimizing the influence of imbalanced class distribution and reducing the training errors caused by overfitting, it is inevitable that a bias is introduced during the label conversion process. Hence, one of our future research directions is to acquire large genuine Email datasets with reliable labels. Topic involved neural network model is proved to be effective for Email document sentiment classification, however, with all sorts of input features, only single sentiment label for each document is a visualized output. Therefore, the other future direction is to explore a more explainable model with richer visual supports.

Declaration of competing interest

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.

CRediT authorship contribution statement

Sisi Liu: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization . **Kyungmi Lee:** Writing - review & editing, Supervision. **Ickjai Lee:** Writing - review & editing, Formal analysis, Supervision, Project administration.

References

[1] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Hum. Lang. Technol. 5 (1) (2012) 1–167.
[2] B. Pang, L. Lee, et al., Opinion mining and sentiment analysis, Foundations and Trends® in Information Retrieval 2 (1–2) (2008) 1–135.
[3] P. Bhatia, Y. Ji, J. Eisenstein, Better document-level sentiment analysis from rst discourse parsing, 2015, arXiv preprint arXiv:1509.01599.
[4] H. Chen, M. Sun, C. Tu, Y. Lin, Z. Liu, Neural sentiment classification with user and product attention, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1650–1659.
[5] S. Ruder, P. Ghaffari, J.G. Breslin, A hierarchical model of reviews for aspect-based sentiment analysis, 2016, arXiv preprint arXiv:1609.02745.
[6] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.
[7] Y. Yin, Y. Song, M. Zhang, Document-level multi-aspect sentiment classification as machine comprehension, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 2044–2054.
[8] S. Liu, I. Lee, Discovering sentiment sequence within email data through trajectory representation, Expert Syst. Appl. 99 (2018) 1–11.
[9] J. Koven, E. Bertini, L. Dubois, N. Memon, Invest: Intelligent visual email search and triage, Digit. Investig. 18 (2016) S138–S148.
[10] J. Tang, H. Li, Y. Cao, Z. Tang, Email data cleaning, in: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, ACM, 2005, pp. 489–498.
[11] S. Poria, I. Chaturvedi, E. Cambria, F. Bisio, Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis, in: 2016 International Joint Conference on Neural Networks, IJCNN, IEEE, 2016, pp. 4465–4473.
[12] N. Chhaya, K. Chawla, T. Goyal, P. Chanda, J. Singh, Frustrated, polite, or formal: Quantifying feelings and tone in email, in: Proceedings of the Second Workshop on Computational Modeling of People'S Opinions, Personality, and Emotions in Social Media, 2018, pp. 76–86.
[13] N. Gupta, M. Gilbert, G.D. Fabbrizio, Emotion detection in email customer care, Comput. Intell. 29 (3) (2013) 489–505.
[14] S. Hangal, M.S. Lam, J. Heer, Muse: Reviving memories using email archives, in: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, ACM, 2011, pp. 75–84.
[15] S.M. Mohammad, T.W. Yang, Tracking sentiment in mail: How genders differ on emotional axes, in: Proceedings of the 2nd Workshop on Computational Approaches To Subjectivity and Sentiment Analysis, Association for Computational Linguistics, 2011, pp. 70–79.

[16] J. Shen, O. Brdiczka, J. Liu, Understanding email writers: Personality prediction from email messages, in: International Conference on User Modeling, Adaptation, and Personalization, Springer, 2013, pp. 318–330.
[17] E. Ezpeleta, U. Zurutuza, J.M.G. Hidalgo, Does sentiment analysis help in bayesian spam filtering? in: International Conference on Hybrid Artificial Intelligence Systems, Springer, 2016, pp. 79–90.
[18] D. Tang, B. Qin, T. Liu, Deep learning for sentiment analysis: Successful approaches and future challenges, Wiley Int. Rev. Data Min. and Knowl. Disc. 5 (6) (2015) 292–303, http://dx.doi.org/10.1002/widm.1171.
[19] L. Zhang, S. Wang, B. Liu, Deep learning for sentiment analysis: A survey, WIREs Data Min. Knowl. Discov. 8 (4) (2018) e1253, http://dx.doi.org/10.1002/widm.1253.
[20] A. Onan, S. Korukoglu, H. Bulut, LDA-based topic modelling in text sentiment classification: An empirical analysis., Int. J. Comput. Linguist. Appl. 7 (1) (2016) 101–119.
[21] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.
[22] J. Goyvaerts, S. Levithan, Regular Expressions Cookbook, O'reilly, 2012.
[23] J. Perkins, Python 3 Text Processing with NLTK 3 Cookbook, Packt Publishing Ltd, 2014.
[24] Y. Wu, L. Liu, C. Pu, W. Cao, S. Sahin, W. Wei, Q. Zhang, A comparative measurement study of deep learning as a service framework, 2018, CoRR abs/1810.12210 arXiv:1810.12210.
[25] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, P.J. Kennedy, Training deep neural networks on imbalanced data sets, in: 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 4368–4374, http://dx.doi.org/10.1109/IJCNN.2016.7727770.
[26] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: Advances in Neural Information Processing Systems, 2015, pp. 649–657.
[27] W.Y. Wang, D. Yang, That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2557–2563.
[28] J.W. Wei, K. Zou, Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019, arXiv preprint arXiv:1901.11196.
[29] G.A. Miller, Wordnet: A lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[30] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[31] R. Řehůřek, P. Sojka, Software framework for topic modelling with large corpora, in: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, ELRA, Valletta, Malta, 2010, pp. 45–50.
[32] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw. 18 (5–6) (2005) 602–610.
[33] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[34] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[35] J. Ulrich, G. Murray, G. Carenini, A publicly available annotated corpus for supervised email summarization, in: Proceedings of AAAI Workshop on Enhanced Messaging, 2008, pp. 77–82.
[36] S. Scott, S. Matwin, Feature engineering for text classification, in: Proceedings of the 16th International Conference on Machine Learning, vol. 99, 1999, pp. 379–388.
[37] J.C. Dill, C.A. Anderson, Effects of frustration justification on hostile aggression, Aggress. Behav. 21 (5) (1995) 359–369.
[38] J.B. Wade, D.D. Price, R.M. Hamer, S.M. Schwartz, R.P. Hart, An emotional component analysis of chronic pain, Pain 40 (3) (1990) 303–310.
[39] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 27.
[40] S. Clinchant, F. Perronnin, Aggregating continuous word embeddings for information retrieval, in: Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, 2013, pp. 100–109.
[41] M.W. Gardner, S. Dorling, Artificial neural networks (the multilayer perceptron)— A review of applications in the atmospheric sciences, Atmos. Environ. 32 (14–15) (1998) 2627–2636.
[42] Y. Kim, Convolutional neural networks for sentence classification, 2014, arXiv preprint arXiv:1408.5882.