Hierarchical State Recurrent Neural Network for Social Emotion Ranking

Deyu Zhou[a,*], Meng Zhang[a], Yang Yang[a], Yulan He[b]

[a] *School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, China*
[b] *Department of Computer Science, University of Warwick, UK*

Abstract

Text generation with auxiliary attributes, such as topics or sentiments, has made remarkable progress. However, high-quality labeled data is difficult to obtain for the large-scale corpus. Therefore, this paper focuses on social emotion ranking aiming to identify social emotions with different intensities evoked by online documents, which could be potentially beneficial for further controlled text generation. Existing studies often consider each document as an entirety that fail to capture the inner relationship between sentences in a document. In this paper, we propose a novel hierarchical state recurrent neural network for social emotion ranking. A hierarchy mechanism is employed to capture the key hierarchical semantic structure in a document. Moreover, instead of incrementally reading a sequence of words or sentences as in traditional recurrent neural networks, the proposed approach encodes the hidden states of all words or sentences simultaneously at each recurrent step to capture long-range dependencies precisely. Experimental results show that the proposed approach performs remarkably better than the state-of-the-art social emotion ranking approaches and is useful for controlled text generation.

Keywords: Sentiment analysis, Social emotion ranking, Attention mechanism

*Corresponding author

Email addresses: d.zhou@seu.edu.com (Deyu Zhou), m.zhang@seu.edu.com (Meng Zhang), yyang@seu.edu.com (Yang Yang), yulan.he@warwick.ac.uk (Yulan He)

1. Introduction

Text generation has attracted a surge of research interest and achieved remarkable progress in various natural language processing applications, such as summary generation [1, 2], machine translation [3, 4, 5] and dialogue generation [6, 7, 8, 9]. While most text generation methods focus on obtaining discriminative representations to improve the quality of generated content, some hidden factors of the texts, such as tenses, sentiment and topics, are ignored in text generation. However, according to the study [10], considering auxiliary factors in text generation process can help to provide a natural, enjoyable and productive human-computer interaction and improve the user satisfaction. For example, if a computer is aware of the emotional state of a user in the conversation, it can generate reasonable responses to offer assistances to a confused user or cheer up a depressed user, which is more appropriate than simply ignoring the user's affective states as is the case with most text generation methods. Several models constrained with auxiliary attributes, such as topics or sentiment labels, have been proposed to generate realistic and controlled texts [11, 12, 13, 14]. However, high-quality labeled data is difficult to obtain for the large-scale corpus. Therefore, this paper focuses on social emotion ranking, providing an effective way to get insight into public opinions [15] and generate precise sentiment labels on online documents, which could be potentially beneficial for further controlled text generation.

Different from traditional sentiment analysis tasks that focus on the classification of emotions from the perspectives of writers, social emotion ranking focuses on identifying readers' emotional responses with different intensities evoked by online documents such as news articles. An example of a news article crawled from Sina News Society Channel with the readers emotion votes is shown in Figure 1. It can be observed that when reading the news article, readers expressed different emotions with the majority voting for '*Sadness*' and '*Touching*' while few people showed '*Shock*', '*Amusement*', '*Curiosity*' and '*Anger*'. In comparison to the total number of votes received, these labels with

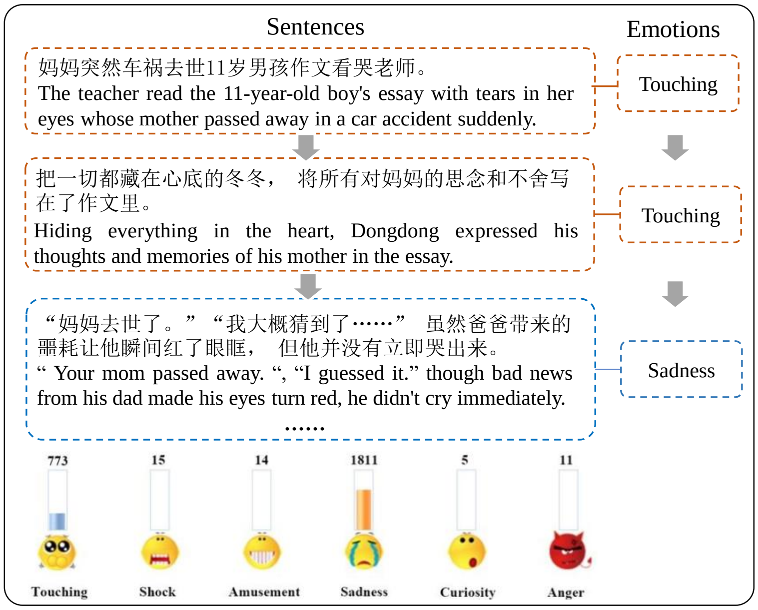

Figure 1: Part of a news article from Sina News Society Channel and its corresponding votes over predefined emotion labels.

few votes could be considered as outliers or irrelevant emotions. Social emotion ranking aims to differentiate relevant emotions from irrelevant ones and only learn the rankings of relevant emotions while neglecting the irrelevant ones.

Approaches for social emotion detection can be categorized into two types, namely, topic-model based methods and discriminative-model based models. Topic-model based methods usually capture topic-emotion information by adding an emotion layer into topic models [16, 17, 18, 19, 20]. However, most methods use the bag-of-word features, which ignore word ordering, or deep semantic representations in the documents and lead to a bottleneck and hinder their use. Discriminative-model based methods for social emotion detection can further be categorized into lexicon-based methods and corpus-based methods. Lexicon-based approaches usually rely on emotion lexicons consisting of emotion words and their corresponding emotion labels to detect emotions from texts [21, 22]. These approaches, which cannot deal with words not in the lexicons, often suffer from low recall. Corpus-based methods aim to train supervised classifiers from annotated training corpus where each document is labeled with emotion class. Emotion detection can be regarded as a single-label classification problem when only choosing the strongest emotion as the label for a given text [23, 24, 25]. It can also be solved using multi-label classification to predict multiple emotions simultaneously [26, 27]. Following this way, a relevant emotion ranking framework was proposed to predict multiple relevant emotions as well as the rankings based on emotion intensities [28, 29].

Recently, deep neural network models have been widely used for text classification. In particular, the attention based recurrent neural networks (RNNs) [30, 31, 32] prevails in text classification. However, RNN-based models are more prone to gradient vanishing due to their sequential nature. An alternative LSTM structure for encoding text was proposed [33] which considers each document as an entity, thus ignores the key hierarchical semantic information of the document.

We argue that the document's hierarchical structure is crucial for expressing the semantic information, which is beneficial for social emotion ranking. In Figure 1, we manually annotate the sentence-level emotions in the right part. It can be seen from the figure that there is emotion transition between neighboring sentences expressing different emotions which together constitute the emotions of the full text. However, existing social emotion ranking approaches usually consider each document as an entirety, hence fail to effectively utilize the document's hierarchical structure and the intrinsic relations between sentences in the semantic meaning of a document. Moreover, long-distance semantic dependencies in texts which are crucial for social emotion ranking are often ignored by the existing methods.

In this paper, we focus on social emotion ranking (SER) by distinguishing multiple relevant social emotions from irrelevant ones and only learn the ranking of relevant social emotions based on their intensities. A novel hierarchical state recurrent neural network (HSRNN) for SER is proposed. The HSRNN encodes the hidden states of all words or sentences simultaneously at each recurrent step rather than incremental reading of the sequences to capture long-range dependencies. Furthermore, a hierarchy mechanism is employed to capture the key hierarchical semantic information of a document, which enables dynamically highlighting important parts in a text evoking the emotions.

The main contributions are summarized below:

- A novel hierarchical state recurrent neural network (HSRNN) is proposed. It incorporates hierarchical state recurrent neural network to capture long-range dependencies and the key semantic hierarchical information of a document.
- Experimental results show that the proposed method performs better than the state-of-the-art emotion ranking methods. Moreover, the important words/sentences highlighted by HSRNN indeed represent the evoked emotions in documents.

The rest of the paper is organized as follows. Section 2 reviews some related work in social emotion detection. Section 3 describes the architecture of the proposed HSRNN. Datasets, evaluating metrics and experimental results are shown in Section 4. Finally, Section 5 concludes the paper and provides directions for future research.

2. Related Work

In general, social emotion detection methods can mainly be categorized into two classes according to their objective functions: topic-model based methods and discriminative-model based methods.

2.1. Topic-model based methods

Topic-model based methods are designed to capture topic-emotion relations by adding an emotion layer into topic models such as Latent Dirichlet Allocation (LDA) [34, 35]. Bao et al. [16] proposed an Emotion-Topic Model (ETM), which first generates a set of topics from different emotions and then generates affective words from each topic. Xu et al. [17] distinguishes the importance of terms and assigns different weights to the terms in the documents according to the result of CHI-test and term frequency, which are further fed into a LDA model. Rao et al. [18] used an Sentiment Latent Topic Mode (SLTM) to associate each topic with social emotions jointly and detect emotion-related and topic-related words. Contextual Sentiment Topic Model (CSTM) [19] assumes that each term is generated from a context-independent topic, a background theme or contextual theme. Tang et al. [20] takes each sentence in the document as an unit and introduces the emotion transition between different sentences into the topic model. However, most topic-model based methods are based on the bag-of-word assumption without considering word ordering and deep semantic meanings, which may lead to a bottleneck and hinder their use.

2.2. Discriminative-model based methods

Discriminative-model based methods for social emotion detection can further be categorized into lexicon-based methods and corpus-based methods. Lexicon-based approaches usually rely on emotion lexicons consisting of emotion words and their corresponding emotion labels for detecting emotions from texts. Many approaches were proposed based on emotion lexicons. For example, Aman and Szpakowicz [36] used the constructed emotion lexicon to classify emotional and non-emotional sentences. Emotion dictionaries could also be constructed from training corpora and then be used to predict the readers' emotion of new articles [21, 22]. Agrawal and An [37] detect emotions from text at sentence level based on contexts. Wang et al. [38] proposed a model with several constraints using non-negative matrix factorization based on an emotion lexicon for multiple emotion detection. Corpus-based methods aim to train supervised classifiers from annotated training corpus where each document is labeled with emotion class. Emotion detection can be regarded as a single-label classification problem when only choosing the strongest emotion as the label for a given text. Strapparava and Mihalcea [23] proposed several knowledge-based and corpus-based methods for emotion classification. Lin, Yang, and

Chen [24] studied the readers' emotion detection with various combinations of feature sets on news articles. Quan et al. [25] proposed a logistic regression model for emotion detection and intermediate hidden variables were also introduced to model the latent structure of input text corpora. Social emotion detection can also be solved using multi-label classification to predict multiple emotions simultaneously. Bhowmick [26] presented a method for classifying news sentences into multiple emotion categories using an ensemble based multi-label classification technique. Zhou et al. [27] proposed a novel approach based on emotion distribution learning to predict multiple emotions with different intensities in a single sentence. Following this way, a relevant label ranking framework for emotion detection was proposed to predict multiple relevant emotions as well as the ranking of emotions based on their intensities [28, 29].

3. Methodology

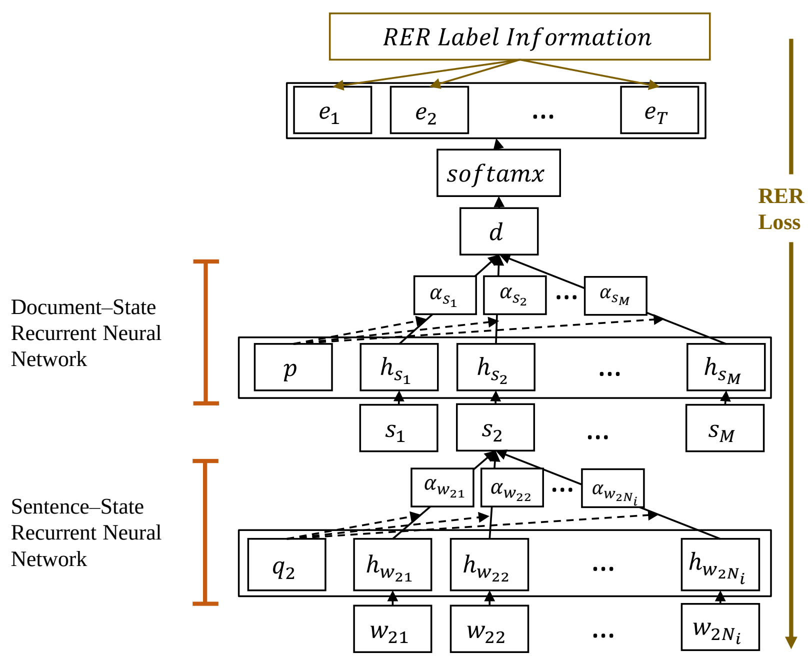

Figure 2: The overall framework of Hierarchical State Recurrent Neural Network (HSRNN).

3.1. Problem setting

Assuming a set of T emotions, $E = \{e_1, e_2, ...e_T\}$, and a set of K document instances, $D = \{d_1, d_2, d_3, ..., d_K\}$, each instance d_i is associated with a ranked list of its relevant emotions $R_i \subseteq E$ and also a list of irrelevant emotions $\overline{R_i} = E - R_i$. Relevant emotion ranking aims to learn a score function $\mathbf{g}(d_i) = [g_1(d_i), ..., g_T(d_i)]$ which assigns a score $g_j(d_i)$ to each emotion $e_j, (j \in \{1, ..., T\})$. The identification of relevant emotions and their rankings can be obtained simultaneously according to their scores assigned by the learned ranking function $\mathbf{g}$.

The learning objective of social emotion ranking (SER) is to both distinguish relevant emotions from irrelevant ones and to rank relevant emotions according to their intensities. Therefore, to fulfil the requirements of SER, the global error function is defined as follows:

$$\mathcal{L} = \sum_{i=1}^{K} \sum_{e_t \in R_i} \sum_{e_s \in \prec(e_t)} \frac{1}{norm_{t,s}} \left[\exp(-(g_t(d_i) - g_s(d_i))) + \omega_{ts}(g_t(d_i) - g_s(d_i))^2 \right] \tag{1}$$

Here, emotion e_s is less relevant than emotion e_t which is represented by $e_s \in \prec (e_t)$. The normalization term $norm_{t,s}$ is used to avoid dominated terms by their set sizes. The term $g_t(d_i) - g_s(d_i)$ measures the difference between two emotions and ω_{ts} represents the relationship between emotion e_t and e_s which is calculated by Pearson correlation coefficient [39].

Figure 2 illustrates the architecture of the hierarchical state recurrent neural network (HSRNN). It consists two main sub-networks: (1) a sentence-state recurrent neural network, which encodes words in the sentence and generate the sentence representation; (2) a document-state recurrent neural network, which encodes sentences in the document to generate the document representation. The document representation is further fed into a softmax layer to obtain relevant labels and their rankings.

3.2. HSRNN encoder

Given a document $d = \{s_1, s_2, ..., s_M\}$, where s_i represents the ith sentence in document d and M is the number of sentences in d. For each sentence, we first map each word in to a fixed word embedding, and the sentence s_i can be represented as $s_i = \{w_{i1}, w_{i2}, ..., w_{iN_i}\}$, where w_{ij} represents the jth word in sentence s_i and N_i is the sentence length.

To encode longer texts, an alternative recurrent neural network [33] is incorporated. For sentence s_i, a state at time step t can be denoted by:

$$H_i^t =< h_{w_{i1}}^t, ..., h_{w_{iN_i}}^t, q_i^t >, \tag{2}$$

which consists of parallel sub states $h_{w_{ij}}^t$ for jth word w_{ij} in sentence s_i and a sentence-level sub state q_i^t.

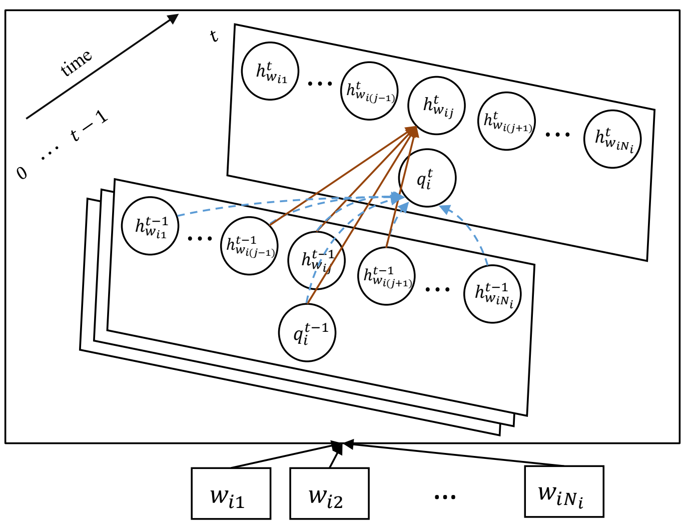

Figure 3: Sentence Encoder.

The recurrent state transition process is used to model information exchange between those sub states, which enriches state representations incrementally. For the initial state H^0, we set $h_{w_{i0}}^0 = q_i^0 = h_0$, where h_0 is a parameter. The state transition is similar to Long Short-Term Memory (LSTM) [40]. And a recurrent cell $c_{w_{ij}}^t$ for each word w_{ij} and a cell $c_{q_i}^t$ for q_i is used. The state transition from H^{t-1} to H^t consists of two parts: (1) the sub-state transitions of each word from $h^{t-1}_{w_{ij}}$ to $h^t_{w_{ij}}$; (2) the sub-state transition of the sentence from q_i^{t-1} to q_i^t. As shown in Figure 3, the value of each $h^t_{w_{ij}}$ is computed based on the values of w_{ij}, $h^{t-1}_{w_{i(j-1)}}$, $h^{t-1}_{w_{ij}}$, $h^{t-1}_{w_{i(j+1)}}$, q_i^{t-1} and their cell values at two adjacent recurrent time steps. The detailed transition is defined as follows:

$$\xi^t_{w_{ij}} = [h^{t-1}_{w_{i(j-1)}}, h^{t-1}_{w_{ij}}, h^{t-1}_{w_{i(j+1)}}] \tag{3}$$

$$\hat{i}^t_{w_{ij}} = \sigma(W_i \xi^t_{w_{ij}} + U_i w_{ij} + V_i q_i^{t-1} + b_i) \tag{4}$$

$$\hat{l}^t_{w_{ij}} = \sigma(W_l \xi^t_{w_{ij}} + U_l w_{ij} + V_l q_i^{t-1} + b_l) \tag{5}$$

$$\hat{r}^t_{w_{ij}} = \sigma(W_r \xi^t_{w_{ij}} + U_r w_{ij} + V_r q_i^{t-1} + b_r) \tag{6}$$

$$\hat{f}^t_{w_{ij}} = \sigma(W_f \xi^t_{w_{ij}} + U_f w_{ij} + V_f q_i^{t-1} + b_f) \tag{7}$$

$$\hat{s}^t_{w_{ij}} = \sigma(W_s \xi^t_{w_{ij}} + U_s w_{ij} + V_s q_i^{t-1} + b_s) \tag{8}$$

$$o^t_{w_{ij}} = \sigma(W_o \xi^t_{w_{ij}} + U_o w_{ij} + V_o q_i^{t-1} + b_o) \tag{9}$$

$$u^t_{w_{ij}} = tanh(W_u \xi^t_{w_{ij}} + U_u w_{ij} + V_u q_i^{t-1} + b_u) \tag{10}$$

$$i^t_{w_{ij}}, l^t_{w_{ij}}, r^t_{w_{ij}}, f^t_{w_{ij}}, s^t_{w_{ij}} = softmax(\hat{i}^t_{w_{ij}}, \hat{l}^t_{w_{ij}}, \hat{r}^t_{w_{ij}}, \hat{f}^t_{w_{ij}}, \hat{s}^t_{w_{ij}}) \tag{11}$$

$$c^t_{w_{ij}} = l^t_{w_{ij}} \odot c^{t-1}_{w_{i(j-1)}} + f^t_{w_{ij}} \odot c^{t-1}_{w_{ij}} + r^t_{w_{ij}} \cdot c^{t-1}_{w_{i(j+1)}} + s^t_{w_{ij}} \cdot c^{t-1}_{q_i} + i^t_{w_{ij}} \cdot u^t_{w_{ij}} \tag{12}$$

$$h^t_{w_{ij}} = o^t_{w_{ij}} \cdot tanh(c^t_{w_{ij}}) \tag{13}$$

where $\xi_{w_{ij}}^t$ is the concatenation of the hidden states of a context window, and the size of the window between two adjacent steps is a hyper parameter that can be set. $i_{w_{ij}}^t, l_{w_{ij}}^t, r_{w_{ij}}^t, f_{w_{ij}}^t$ and $s_{w_{ij}}^t$ are the gates that control the information flow from $c_{w_{i(j-1)}}^{t-1}$, $c_{w_{ij}}^{t-1}$, $c_{w_{i(j+1)}}^{t-1}$, $c_{q_i}^{t-1}$ and $\xi_{w_{ij}}^t$ to $c_{w_{ij}}^t$ respectively. And $o_{w_{ij}}^t$ is an output gate that control the information from the cell state $c_{w_{ij}}^t$ to the hidden state $h_{w_{ij}}^t$. W_x, U_x, V_x and $b_x (x \in \{i, o, l, r, f, s, u\})$ are model parameters.

Similarly, the value of q_i^t is computed based on the values of $h_{w_{ij}^{t-1}}$ as below:

$$\bar{h}_i^{t-1} = avg(h_{w_{i0}^{t-1}}, h_{w_{i1}^{t-1}}, \ldots, h_{w_{iN_i}^{t-1}}) \tag{14}$$

$$\hat{g}_{q_i}^t = \sigma(W_g q_i^{t-1} + U_q \bar{h}_i^{t-1} + b_q) \tag{15}$$

$$\hat{f}_{w_{ij}}^t = \sigma(W_f q_i^{t-1} + U_f h_{w_{ij}}^t + b_w) \tag{16}$$

$$o_{q_i}^t = \sigma(W_o q_i^{t-1} + U_o \bar{h}_i^{t-1} + b_o) \tag{17}$$

$$f_{w_{i1}}^t, f_{w_{i2}}^t, \ldots, f_{w_{iN_i}}^t, g_{q_i}^t = softmax(\hat{f}_{w_{i1}}^t, \hat{f}_{w_{i2}}^t, \ldots, \hat{f}_{w_{iN_i}}^t, \hat{g}_{q_i}^t) \tag{18}$$

$$c_{q_i}^t = g_{q_i}^t \cdot q_i^{t-1} + \sum_j f_{w_{ij}}^t \cdot c_{w_{ij}}^{t-1} \tag{19}$$

$$q_i^t = o_{q_i}^t \cdot tanh(c_{q_i}^t) \tag{20}$$

where $\bar{h}_i^{t-1}$ is the average of the hidden states of the words in the sentence. $g_{q_i}^t$ and $f_{w_{ij}}^t$ are the gates that control the information flow from q_i^{t-1} and $c_{w_{ij}}^{t-1}$ to $c_{q_i}^t$ respectively. And $o_{q_i}^t$ is an output gate that control the information from the cell state $c_{q_i}^t$ to the hidden state q_i^t. W_x, U_x and $b_x (x \in \{q, w, o\})$ are model parameters.

3.3. Sentence-state recurrent neural network

The sentence-state recurrent neural network is employed to encode words in the sentence and generate the sentence representation s_i. Firstly, it feeds the word embedding into the HSRNN encoder to obtain the hidden states H_i consisting of a hidden vector $h_{w_{ij}}$ for each word w_{ij}, and a global sentence-level hidden vector q_i.

$$H_i = [h_{w_{i0}}, h_{w_{i1}}, \ldots, h_{w_{iN_i}}, q_i] = Encoder(w_{i1}, w_{i2}, \ldots, w_{iN_i}) \tag{21}$$

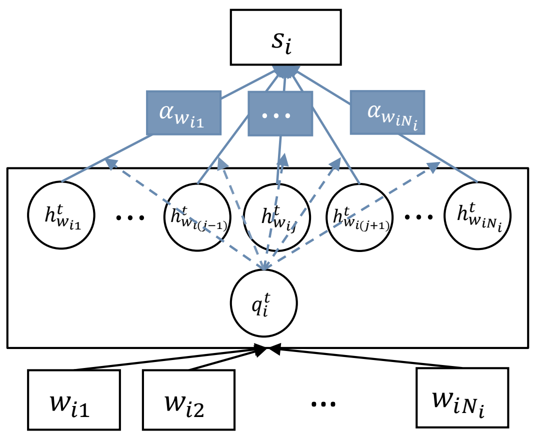

Figure 4: Sentence Encoder with Attention.

Not all words contribute equally to the meaning of a sentence. Therefore, as shown in Figure 4, we further introduce an attention mechanism to extract words with great importance and aggregate the representation of those informative words to form the final sentence representation. More concretely,

$$\varphi_{w_{ij}} = tanh(W_a(h_{w_{ij}} + q_i) + b_a) \tag{22}$$

$$\alpha_{w_{ij}} = \frac{exp(\varphi^{\top}_{w_{ij}} u_a)}{\sum_j exp(\varphi^{\top}_{w_{ij}} u_a)} \tag{23}$$

$$s_i = \sum_j \alpha_{w_{ij}} h_{w_{ij}} \tag{24}$$

where the weight $\alpha_{w_{ij}}$ is the attention of each word w_{ij} and W_a, b_a and u_a are parameters which are similar to [41]. Note that we further incorporate the global information of sentence representation q_i^t to strengthen the attention.

3.4. Document-state recurrent neural network

The document-state recurrent neural network is employed to encode sentences in document d to generate the final document representation. Firstly, it feeds the sentence representations into the encoder to obtain the hidden states consisting of a hidden vector h_{s_i} for each sentence and a global document-level hidden vector p.

$$H = [h_{s_1}, h_{s_2}, \ldots, h_{s_M}, p] = Encoder(s_1, s_2, \ldots, s_M) \tag{25}$$

where h_{s_i} represents syntactic and semantic features for sentence s_i, while p represents features for the whole document.

Since not all sentences contribute equally to the final semantic meaning of the document, an attention mechanism is also used here to capture the relative importance of each sentence contributing to the final document representation d as below:

$$\varphi_{s_i} = tanh(W_b(h_{s_i} + p) + b_b) \tag{26}$$

$$\alpha_{s_i} = \frac{exp(\varphi_{s_i}^\top u_b)}{\sum_j exp(\varphi_{s_i}^\top u_b)} \tag{27}$$

$$d = \sum_j \alpha_{s_i} h_{s_i} \tag{28}$$

where α_{s_i} is the weight of s_i and d is the syntactic and semantic representation for the document. And we also incorporate the global information of document representation p obtained from the encoder to strengthen the attention.

3.5. Social emotion ranking

Algorithm 1 HSRNN model

Require: Document $d = \{s_1, s_2, ..., s_M\}$, where $s_i = \{w_{i1}, w_{i2}, ..., w_{iN_i}\}$, and threshold Θ

return Social emotion ranking result $\hat{L}$

1. <u>Sentence-state recurrent neural network</u>

for s_i in d **do**

  for w_{ij} in s_i **do**

    $h_{w_{ij}} = Encoder(w_{ij})$

  end for

  $s_i = Attention(h_{w_{i1}}, h_{w_{i2}}, \ldots, h_{w_{iN_i}}, q_i)$

end for

2. <u>Document-state recurrent neural network</u>

for s_i in d **do**

  $h_{s_i} = Encoder(s_i)$

end for

$d = Attention(h_{s_1}, h_{s_2}, \ldots, h_{s_M}, p)$

3. <u>Social emotion ranking</u>

$\hat{y} = softmax(W_y d + b_y)$

Let $L = \varnothing$

for $\hat{y}_i$ in $\hat{y}$ **do**

  if $\hat{y}_i > \Theta$ **then**

    Add $\hat{y}_i$ to L

  end if

end for

$\hat{L} = sort(L)$

After obtaining the document representation d, we use a linear layer to transform d into a label vector, which is further fed into a softmax layer to calculate the predicted scores of different emotions.

$$\hat{y} = softmax(W_y d + b_y) \tag{29}$$

where W_y and b_y are model parameters.

In order to differentiate relevant emotions from irrelevant ones, we need to define a threshold Θ which could be simply set to a fixed value or learned from data [42]. Those emotions with scores lower than the threshold will be considered as irrelevant and hence discarded. Therefore, we can get the relevant emotion labels $L = \{l_1, l_2, \ldots, l_Q\}$ of d and sort them to get the final ranking result $\hat{L}$.

The whole procedure of HSRNN is summarized in Algorithm 1.

4. Experiments

To evaluate the effectiveness of the proposed approach, we conducted experiments on two real-world corpora.

- **Sina Social News (News)** [28] was collected from the Sina news *Society* channel where readers can choose one of the six emotions such as *Amusement*, *Touching*, *Anger*, *Sadness*, *Curiosity*, and *Shock* after reading a news article. In total, 5,586 news articles published from January 2014 to July 2016 were kept, together with the readers' emotion votes. The statistics of the News corpus are shown in Table 1.

- **Ren-CECps corpus (Blogs)** [43] contains 1,487 blogs. Each blog in the dataset is annotated with eight basic emotions, including *Anger*, *Anxiety*, *Expect*, *Hate*, *Joy*, *Love*, *Sorrow* and *Surprise*, which are represented by their emotion intensities in the range of [0, 1]. The statistics of the Blog corpus are shown in Table 2.

Category	Touching	Shock	Amusement	Sadness	Curiosity	Anger
#Votes	694,006	572,651	869,464	837,431	212,559	1,109,315

Table 1: Statistics for the News corpus used in our experiments.

Category	Anger	Anxiety	Expect	Hate	Joy	Love	Sorrow	Surprise
#Scores	116.4	422.6	385.5	174.2	349.2	610.6	408.4	59.2

Table 2: Statistics for the Blog corpus used in our experiments.

Documents were preprocessed with the python jieba segmenter[1] for word segmentation and filtering. In our experiments, we set the word embedding dimension to 300 and train word embeddings with GloVe[2]. The threshold Θ was set to 0.1. The hyper-parameters were chosen empirically on the validation set. For each method, 10-fold cross validation is conducted using the same feature construction method to get the final performance. Evaluation metrics typically used in multi-label and label ranking are employed [44]. The definition of evaluation metrics [45, 46, 47, 48, 49, 50] are defined as follows.

$$ProLoss = \frac{1}{n}\sum_{i=1}^{n}\sum_{e_t \in R_i \cup \{\Theta\}}\sum_{e_s \in \prec(e_t)} \frac{1}{norm_{t,s}} l_{t,s} \tag{30}$$

where $l_{t,s}$ is a modified 0-1 error and $norm_{t,s}$is the set size of label pair(e_t, e_s).

$$HammingLoss = \frac{1}{nT}\sum_{i=1}^{n} |\hat{R}_i \triangle R_i| \tag{31}$$

where $\hat{R}_i$ is the predicted relevant emotions.

$$RankingLoss = \frac{1}{n}\sum_{i=1}^{n} \frac{(\sum_{(e_t,e_s)\in R_i \times \overline{R_i}} \delta[g_t(x_i) < g_s(x_i)])}{(|R_i| \times |\overline{R_i}|)} \tag{32}$$

where $\hat{R}_i$ is the predicted relevant emotions.

$$OneError = \frac{1}{n}\sum_{i=1}^{n} \delta[\operatorname*{argmax}_{\mathbf{e_t}} g_t(x_i) \notin R_i] \tag{33}$$

$$AveragePrecision = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|R_i|} \times \sum_{t:e_t \in R_i} \frac{|\{e_s \in R_i | g_s(x_i) > g_t(x_i)\}|}{|\{e_s | g_s(x_i) > g_t(x_i)\}|} \tag{34}$$

[1]`https://github.com/fxsjy/jieba`
[2]`https://nlp.stanford.edu/projects/glove/`

$$Coverage = \frac{1}{n}\sum_{i=1}^{n}\max_{t:e_t\in R_i}|\{e_s|g_s(x_i) > g_t(x_i)\}| \quad (35)$$

$$F1_{exam} = \frac{1}{n}\sum_{i=1}^{n}\frac{2|R_i \cap \hat{R}_i|}{(|R_i| + |\hat{R}_i|)} \quad (36)$$

$$MicroF1 = F1(\sum_{t=1}^{T}TP_t, \sum_{t=1}^{T}FP_t, \sum_{t=1}^{T}TN_t, \sum_{t=1}^{T}FN_t) \quad (37)$$

where TP_t, FP_t, TN_t and FN_t represent the number of true positive, false positive, true negative, and false negative test examples with respect to emotion t respectively. $F1(TP_t,\ FP_t, TN_t, FN_t)$ represents the specific binary classification metric F1 [51].

$$MacroF1 = \frac{1}{T}\sum_{t=1}^{T}F1(TP_t, FP_t, TN_t, FN_t) \quad (38)$$

Note that metrics from $ProLoss$ to $F1_{exam}$ work by evaluating performance on each test example separately and returning the mean value across test set. $MicroF1$ and $MacroF1$ work by evaluating performance on each emotion category separately and returning the macro/micro-averaged value across all emotion categories.

4.1. Comparison with emotion ranking methods

There are several approaches addressing social emotions ranking from texts. We compared HSRNN with some state-of-the-art emotion ranking methods.

- **EmoDetect** [38] outputs emotion distribution using non-negative matrix factorization which combines several constraints.
- **EDL** [27] learns a mapping function from text to emotion distribution based on label distribution learning.
- **RER** [28] performs relevant emotion ranking using SVMs.

Method	PL(↓)	HL(↓)	RL(↓)	OE(↓)	AP(↑)	Cov(↓)	F1(↑)	MiF1(↑)	MaF1(↑)
EDL	0.2348	0.2510	0.1616	0.2243	0.8372	2.1940	0.6260	0.6454	0.5703
EmoDetect	0.2157	0.2575	0.1538	0.1627	0.8605	2.1761	0.6697	0.6739	0.5359
RER	0.2142	0.2498	0.1491	0.1513	0.8633	2.1989	0.6820	0.6919	0.6198
INN-RER	0.1973	0.2312	0.1353	0.1331	0.8764	2.1339	0.7108	0.7161	0.6282
HSRNN	**0.1766**	**0.1909**	**0.1133**	**0.1076**	**0.8966**	**1.9225**	**0.7449**	**0.7313**	**0.6326**

Table 3: Experimental results of the proposed approach and the baselines on News corpus. 'PL' represent Pro Loss, 'HL' represents Hamming Loss, 'RL' represents ranking loss, 'OE' represents one error, 'AP' represent average precision, 'Cov' represent coverage, 'F1' represents $F1_{exam}$, 'MiF1' represents MicroF1, 'MaF1' represents MacroF1. "↓" indicates "the smaller the better", while "↑" indicates "the larger the better". The best performance on each evaluation measure is highlighted by boldface.

Method	PL(↓)	HL(↓)	RL(↓)	OE(↓)	AP(↑)	Cov(↓)	F1(↑)	MiF1(↑)	MaF1(↑)
EDL	0.3385	0.3916	0.2550	0.4206	0.6962	4.2491	0.5060	0.5396	0.4131
EmoDetect	0.3115	0.3848	0.2123	0.2880	0.7617	4.1650	0.5340	0.5492	0.4387
RER	0.3007	0.3657	0.2043	0.2728	0.7746	4.1638	0.5957	0.6084	**0.5342**
INN-RER	0.2829	0.3209	0.1924	0.2626	0.7784	3.6418	**0.6187**	**0.6225**	0.5133
HSRNN	**0.2676**	**0.2774**	**0.1573**	**0.2023**	**0.8049**	**3.4935**	0.5829	0.5736	0.4637

Table 4: Experimental results of the proposed approach and the baselines on Blog corpus. The best performance on each evaluation measure is highlighted by boldface.

- **INN-RER** [29] performs relevant emotion ranking using a three-layer neural network combined with a topic model.

Experimental results on News corpus and Blog corpus are summarized in Table 3 and Table 4 respectively. It can be observed that: (1) HSRNN outperforms the baselines on almost evaluation metrics on the two corpora. It verifies the effectiveness of HSRNN, which can capture long-distance dependencies and the key hierarchical semantic meaning in the document; (2) On the Blog corpus, INN-RER and RER work better than HSRNN with multi-label evaluation metrics $F1$, $MicroF1$ and $MacroF1$, but worse than HSRNN on the ranking evaluation metrics especially $ProLoss$, $HammingLoss$ and $RankingLoss$. It indicates that when addressing social emotion problems, INN-RER and RER are more suitable to differentiate relevant emotions from irrelevant ones but fail to give the proper rankings of relevant emotions while HSRNN can capture the ranking information between different labels and is much better for social emotion ranking task.

4.2. Comparison with multi-label methods

Since SER can be treated as an extension of multi-label problem, so we also compare HSRNN with some widely-used multi-label methods.

- **BP-MLL** [52] employs a novel error function into back propagation algorithm to capture the characteristics of multi-label learning.
- **ML-KNN** [53] utilizes maximum a posteriori (MAP) principle in the K-nearest neighbor (KNN) algorithm.
- **ML-RBF** [54] uses radial basis function (RBF) to solve multi-label problem.
- **ECC** [55] applies classifier chains in an ensemble framework.
- **LIFT** [56] constructs label-specific features to deal with multi-label problem.
- **MLLOC** [57] exploits local emotion correlations in expression data.
- **Rank-SVM** [58] uses a large margin strategy to distinguish relevant labels from irrelevant ones.

Experimental results on News corpus are shown in Table 5. It can be seen from Table 5 that HSRNN works much better than the baselines on all evaluation metrics. These multi-label methods often use bag-of-words (BOW) or TF-IDF as inputs without word ordering, which limits making full use of the sequential context information of documents. However, HSRNN utilizes a hierarchical architecture to obtain the semantic representation of the document, which can capture the key hierarchical semantic structure of a document and is able to attend to the most important words/sentences that evoke emotions.

Method	PL(↓)	HL(↓)	RL(↓)	OE(↓)	AP(↑)	Cov(↓)	F1(↑)	MiF1(↑)	MaF1(↑)
BP-MLL	0.2118	0.2399	0.1443	0.1544	0.8677	2.1738	0.6881	0.6915	0.6013
ML-KNN	0.2863	0.4415	0.1780	0.2079	0.8261	2.2204	0.6046	0.6220	0.5396
ML-RBF	0.2575	0.4164	0.1283	0.1290	0.8796	2.0143	0.6261	0.6379	0.5746
ECC	0.2095	0.2428	0.1464	0.1272	0.8598	2.0948	0.6876	0.6923	0.6130
LIFT	0.2224	0.3363	0.1382	0.1411	0.8234	2.1394	0.6646	0.6801	0.6151
MLLOC	0.4458	0.4206	0.4500	0.4193	0.6531	3.9032	0.3000	0.4060	0.3327
Rank-SVM	0.2842	0.2872	0.2114	0.2034	0.7967	2.5358	0.5066	0.5656	0.5298
HSRNN	**0.1766**	**0.1909**	**0.1133**	**0.1076**	**0.8966**	**1.9225**	**0.7449**	**0.7313**	**0.6326**

Table 5: Comparison with multi-label methods on News corpus.

Method	PL(↓)	HL(↓)	RL(↓)	OE(↓)	AP(↑)	Cov(↓)	F1(↑)	MiF1(↑)	MaF1(↑)
S-LSTM+ATT	0.2010	0.2356	0.1440	0.1082	0.8860	2.3143	0.7257	0.7194	0.5738
HAN	0.2021	0.2313	0.1415	0.1100	0.8840	2.3124	0.7093	0.7034	0.6006
HSRNN	**0.1766**	**0.1909**	**0.1133**	**0.1076**	**0.8966**	**1.9225**	**0.7449**	**0.7313**	**0.6326**

Table 6: Experimental results of HSRNN and two sub-networks on News corpus.

4.3. Model analysis

In order to analyze the contributions of different components of HSRNN, we compare HSRNN with two sub-networks.

- **S-LSTM + ATT** [33] regards the document as an entirety. The attention mechanism is also incorporated.
- **HAN** [31] employs a hierarchical attention network for document classification with traditional recurrent neural network as encoder.

Experimental results are summarized in Table 6. It can be observed from Table 6 that: (1) HSRNN achieves better performance than S-LSTM+ATT, which verifies the effectiveness of employing the key hierarchical semantic structure in a document; (2) HSRNN performs remarkably better than HAN, which further verifies the effectiveness of HSRNN in capturing long-range dependencies.

4.4. Visualization of hierarchical attention weights

To further investigate whether HSRNN is able to capture the key hierarchical semantic structure which is more important for revealing the emotions expressed in texts, we visualize the hierarchical attention weights for an example document in Figure 5 and Figure 6. Each row is a sentence with blue color with varying intensities indicating word importance scores. The leftmost vertical brown color bar indicates varying sentence importance scores. Darker color means more important. As the document is too long, we manually simplify the text for a better visualization and provide an English translation of each sentence.

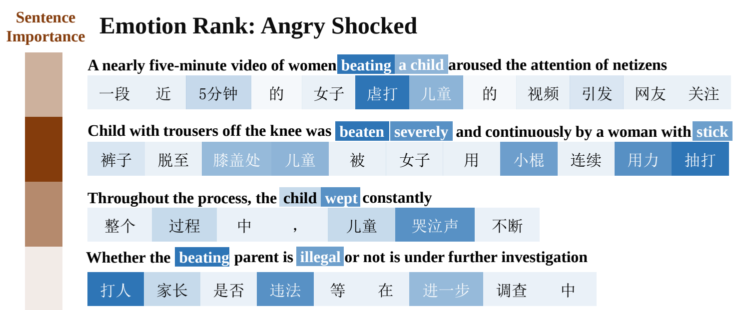

Figure 5: Attention visualization of Case 1.

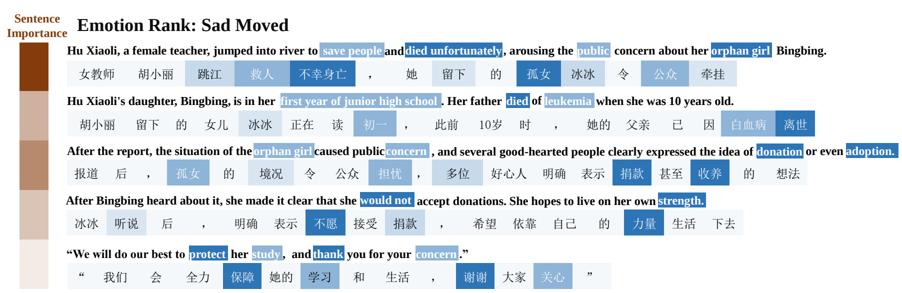

Figure 6: Attention visualization of Case 2.

It can be observed that HSRNN can figure out both important words and sentences which contribute to the most of the emotions associated with text. For example, In Figure 5, words including '*severely*' and '*beaten*' and their corresponding sentences are highlighted which evoke the emotions of *Angry* and *Shocked*. In Figure 6, HSRNN assigned larger weights to words like '*saved people*', '*unfortunately*', '*orphan girl*' and '*adopted*' and their corresponding sentences are highlighted which evokes the emotions of *Moved* and *Sad*.

4.5. Visualization of text generation

		Text	Emotion
S1	Original	太**可怜**了，老妈！ What a **pity**, mom!	Sorrow
	Generated	你**开心**了，真**好**！ It's so **nice** that you are **happy**!	Joy
S2	Original	坐久了，心也**凉**了，这就是海边的**孤独**。 After sitting for a long time, my heart is **cold**. This is the **loneliness** of the seaside.	Sorrow
	Generated	坐久了，心也**安**了，这就是心灵的**快乐**。 After sitting for a long time, the heart is **at ease**. This is the **happiness** of the soul.	Joy
S3	Original	一路我是**笑**着回去的。 I went back **laughing** all the way.	Joy
	Generated	于是我表情是**悲伤**的。 So my expression was **sad**.	Sorrow
S4	Original	如此的**激情**，如此的**兴奋**，又如此的羞涩。 So **passionate**, so **excited**, and so shy.	Joy
	Generated	**寂寞**的夜，**寂寞**的声音，又如时间的**哀愁**。 The **lonely** night and the **lonely** voice are like the **sorrow** of time.	Sorrow
S5	Original	这是我有生以来过的最**开心**的节日。 This is the **happiest** festival I've ever had.	Joy
	Generated	这是我出生以来去的最**痛苦**的地方。 This is the most **painful** place I've been to since I was born.	Sorrow

Figure 7: Generated texts controlled by social emotions. Each pair of sentences consists of original sentences and generated sentences. The emotion labels of original texts are obtained from HSRNN.

To further investigate the influence of sentiment labels in text generation process, a text generation model constrained with auxiliary attributes[3] is utilized to generate texts controlled by social emotions. The input of the model is Blog texts and the corresponding most relevant emotions obtained

[3] `https://github.com/asyml/texar/tree/master/examples/text_style_transfer`

from HSRNN and the output is the generated content with opposite emotions. Note that the opposite emotions are not fed into the text generation model, but to simplify the problem, we only choose the texts with *Joy* or *Sorrow* emotions. Figure 7 presents some examples generated by the model. The words marked in red are the emotion words corresponding to the emotion category. For better visualization results, we provide English translations of each sentence. It can be seen from the figure that text generation model can generate diverse controlled texts under the guidance of the auxiliary emotion labels. For example, the original text in S1 shows *Sorrow* emotion with the word '*pity*' while the generated content with the words '*nice*' and '*happy*' reflects *Joy* emotion. Therefore, with the help of the auxiliary emotion labels, text generation model can generate diverse and interpretable controlled texts, demonstrating the benefit of sentiment analysis model in text generation process.

5. Conclusion

Social emotion ranking provides an effective way to generate precise sentiment labels on online documents, which could be potentially beneficial for further controlled text generation. In this paper, we have proposed a novel hierarchical state recurrent neural network for social emotion ranking which could be potentially useful for further controlled text generation. Instead of incrementally reading a sequence of words, this model encodes the hidden states of all words/sentences simultaneously which can capture long-range dependencies. Moreover, this model captures the key hierarchical semantic structure of a document and is able to attend to the most important words/sentences that evoke emotions. Experimental results show that the proposed approach performs remarkably better than the state-of-the-art emotion ranking approaches. In the future, we will explore learning emotions events and social emotion ranking function simultaneously in a unified framework.

Acknowledgements

We are grateful to the reviewers for their valuable comments and constructive suggestions. This work was funded by the National Key Research and Development Program of China (2016YFC1306704), the National Natural Science Foundation of China (61772132), Innovate UK (grant no. 103652), the EPSRC (grant no. EP/T017112/1, EP/V048597/1) and a Turing AI Fellowship funded by the UK Research and Innovation (UKRI) (grant no. EP/V020579/1)..

[1] S. Takase, J. Suzuki, N. Okazaki, T. Hirao, M. Nagata, Neural headline generation on abstract meaning representation, in: Proceedings of the 2016 conference on empirical methods in natural language processing, 2016, pp. 1054–1059.

[2] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al., Abstractive text summarization using sequence-to-sequence rnns and beyond, arXiv preprint arXiv:1602.06023.

[3] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473.

[4] M. Artetxe, G. Labaka, E. Agirre, K. Cho, Unsupervised neural machine translation, arXiv preprint arXiv:1710.11041.

[5] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, Ł. Kaiser, N. Kalchbrenner, N. Parmar, et al., Tensor2tensor for neural machine translation, arXiv preprint arXiv:1803.07416.

[6] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, D. Jurafsky, Adversarial learning for neural dialogue generation, arXiv preprint arXiv:1701.06547.

[7] H. Chen, Z. Ren, J. Tang, Y. E. Zhao, D. Yin, Hierarchical variational memory network for dialogue generation, in: Proceedings of the 2018 World Wide Web Conference, 2018, pp. 1653–1662.

[8] R. Li, X. Li, C. Lin, M. Collinson, R. Mao, A stable variational autoencoder for text modelling, in: Proceedings of the 12th International Conference on Natural Language Generation (INLG), 2019.

[9] R. Li, X. Li, G. Chen, C. Lin, Improving variational autoencoder for text modelling with timestep-wise regularisation, Proceedings of the 28th International Conference on Computational Linguistics (COLING).

[10] H. Prendinger, J. Mori, M. Ishizuka, Using human physiology to evaluate subtle expressivity of a virtual quizmaster in a mathematical game, International Journal of Human-Computer Studies 62 (2) (2005) 231–245.

[11] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, E. P. Xing, Toward controlled generation of text, arXiv preprint arXiv:1703.00955.

[12] H. Zhou, M. Huang, T. Zhang, X. Zhu, B. Liu, Emotional chatting machine: Emotional conversation generation with internal and external memory, arXiv preprint arXiv:1704.01074.

[13] W. Wang, Z. Gan, H. Xu, R. Zhang, G. Wang, D. Shen, C. Chen, L. Carin, Topic-guided variational autoencoders for text generation, arXiv preprint arXiv:1903.07137.

[14] X. Li, C. Lin, R. Li, C. Wang, F. Guerin, Latent space factorisation and manipulation via matrix subspace projection, in: International Conference on Machine Learning, PMLR, 2020, pp. 5916–5926.

[15] R. W. Picard, R. Picard, Affective computing, Vol. 252, MIT press Cambridge, 1997.

[16] S. Bao, S. Xu, L. Zhang, R. Yan, Z. Su, D. Han, Y. Yu, Joint emotion-topic modeling for social affective text mining, in: 2009 Ninth IEEE International Conference on Data Mining, IEEE, 2009, pp. 699–704.

[17] R. Xu, L. Ye, J. Xu, Readers emotion prediction based on weighted latent dirichlet allocation and multi-label k-nearest neighbor model, Journal of Computational Information Systems 9 (6) (2013) 2209–2216.

[18] Y. Rao, Q. Li, X. Mao, W. Liu, Sentiment topic models for social emotion mining, Information Sciences 266 (5) (2014) 90–100.

[19] Y. Rao, Contextual sentiment topic model for adaptive social emotion classification, IEEE Intelligent Systems 31 (1) (2016) 41–47.

[20] D. Tang, Z. Zhang, Y. He, C. Lin, D. Zhou, Hidden topicemotion transition model for multi-level social emotion detection, Knowledge Based Systems 164 (2019) 426–435.

[21] Y. Rao, X. Quan, L. Wenyin, Q. Li, M. Chen, Building word-emotion mapping dictionary for online news, in: SDAD 2012 The 1st International Workshop on Sentiment Discovery from Affective Data, 2012, p. 28.

[22] J. Lei, Y. Rao, Q. Li, X. Quan, L. Wenyin, Towards building a social emotion detection system for online news, Future Generation Computer Systems 37 (2014) 438–448.

[23] C. Strapparava, R. Mihalcea, Learning to identify emotions in text, in: Proceedings of the 2008 ACM symposium on Applied computing, ACM, 2008, pp. 1556–1560.

[24] K. H.-Y. Lin, C. Yang, H.-H. Chen, Emotion classification of online news articles from the reader's perspective, in: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, IEEE Computer Society, 2008, pp. 220–226.

[25] X. Quan, Q. Wang, Y. Zhang, L. Si, L. Wenyin, Latent discriminative models for social emotion detection with emotional dependency, ACM Trans. Inf. Syst. 34 (1) (2015) 2:1–2:19.

[26] P. K. Bhowmick, Reader perspective emotion analysis in text through ensemble based multi-label classification framework, Computer and Information Science 2 (4) (2009) 64.

[27] D. Zhou, X. Zhang, Y. Zhou, Q. Zhao, X. Geng, Emotion distribution learning from texts, in: Conference on Empirical Methods in Natural Language Processing, 2016, pp. 638–647.

[28] D. Zhou, Y. Yang, Y. He, Relevant emotion ranking from text constrained with emotion relationships, in: Meeting of the North American Chapter of the Association for Computation Linguistics, 2018, pp. 561–571.

[29] Y. Yang, D. Zhou, Y. He, An interpretable neural network with topical information for relevant emotion ranking, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, 2018, pp. 3423–3432.
URL `https://aclanthology.info/papers/D18-1379/d18-1379`

[30] M. Schuster, K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (11) (2002) 2673–2681.

[31] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, 2016, pp. 1480–1489. `doi:10.18653/v1/N16-1174`.

[32] R. Li, C. Lin, M. Collinson, X. Li, G. Chen, A dual-attention hierarchical recurrent neural network for dialogue act classification, in: Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), 2019, pp. 383–392.

[33] Y. Zhang, Q. Liu, L. Song, Sentence-state lstm for text representation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2018, pp. 317–327.
URL `http://aclweb.org/anthology/P18-1030`

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

[35] Y. He, C. Lin, A. E. Cano, Online sentiment and topic dynamics tracking over the streaming data, in: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, IEEE, 2012, pp. 258–266.

[36] S. Aman, S. Szpakowicz, Identifying expressions of emotion in text, Lecture Notes in Computer Science 4629 (2007) 196–205.

[37] A. Agrawal, A. An, Unsupervised emotion detection from text using semantic and syntactic relations, in: Ieee/wic/acm International Conferences on Web Intelligence and Intelligent Agent Technology, 2012, pp. 346–353.

[38] Y. Wang, A. Pal, Detecting emotions in social media: A constrained optimization approach, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), 2015, pp. 996–1002.

[39] W. A. Nicewander, Thirteen ways to look at the correlation coefficient, American Statistician 42 (1) (1988) 59–66.

[40] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.

[41] N. Pappas, A. Popescu-Belis, Multilingual hierarchical attention networks for document classification, CoRR abs/1707.00896. `arXiv:1707.00896`. URL `http://arxiv.org/abs/1707.00896`

[42] E. L. Mencia, J. Furnkranz, Pairwise learning of multilabel classifications with perceptrons (2008) 2899–2906.

[43] M. Quan, O. M. Tepper, K. Small, A. Fadl, C. Quan, N. P. G. Kumar, M. Choi, N. Karp, F. Ren, Sentence emotion analysis and recognition based on emotion words using ren-cecps , 2010.

[44] F. Sebastiani, Machine learning in automated text categorization, Acm Computing Surveys 34 (1) (2001) 1–47.

[45] R. E. Schapire, Y. Singer, A boosting-based system for text categorization, Machine Learning 39 (2-3) (2000) 135–168.

[46] A. Elisseeff, J. Weston, Kernel methods for multi-labelled classification and categ orical regression problems.

[47] M. L. Zhang, Z. H. Zhou, Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038–2048.

[48] Y. Kai, S. Yu, V. Tresp, Multi-label informed latent semantic indexing, in: International Acm Sigir Conference on Research and Development in Information Retrieval, 2005.

[49] S.Godbole, S.Sarawagi, Discriminative methods for multi-labeled classification, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2004, pp. 22–30.

[50] Y. Yang, An evaluation of statistical approaches to text categorization, Proc Amia Annu Fall Symp 1 (1-2) (1999) 358–362.

[51] C. D. Manning, P. Raghavan, H. Schtze, An introduction to information retrieval, Journal of the American Society for Information Science and Technology 43 (3) (2008) 824–825.

[52] M. L. Zhang, Z. H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge Data Engineering 18 (10) (2006) 1338–1351.

[53] M. L. Zhang, Z. H. Zhou, Ml-knn: A lazy learning approach to multi-label learning, Pattern Recognition 40 (7) (2007) 2038–2048.

[54] M. L. Zhang, Ml-rbf : Rbf neural networks for multi-label learning, Neural Processing Letters 29 (2) (2009) 61–74.

[55] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2009, pp. 254–269.

[56] M. L. Zhang, Lift: multi-label learning with label-specific features, in: International Joint Conference on Artificial Intelligence, 2011, pp. 1609–1614.

[57] S. J. Huang, Z. H. Zhou, Multi-label learning by exploiting label correlations locally, in: Twenty-Sixth AAAI Conference on Artificial Intelligence, 2012, pp. 949–955.

[58] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE transactions on knowledge and data engineering 26 (8) (2014) 1819–1837.