Encoding Syntactic Dependency and Topical Information for Social Emotion Classification

Chang Wang, Bang Wang, Wei Xiang and Minghua Xu*
Huazhong University of Science and Technology (HUST), Wuhan, China
wang_chang,wangbang,xiangwei,xuminghua@hust.edu.cn

ABSTRACT

Social emotion classification is to estimate the distribution of readers' emotion evoked by an article. In this paper, we design a new neural network model by encoding sentence syntactic dependency and document topical information into the document representation. We first use a dependency embedded recursive neural network to learn syntactic features for each sentence, and then use a gated recurrent unit to transform the sentences' vectors into a document vector. We also use a multi-layer perceptron to encode the topical information of a document into a topic vector. Finally, a gate layer is used to compose the document representation from the gated summation of the document vector and the topic vector. Experiment results on two public datasets indicate that our proposed model outperforms the state-of-the-art methods in terms of better average Pearson correlation coefficient and MicroF1 performance.

CCS CONCEPTS

• **Information systems** → **Sentiment analysis**.

KEYWORDS

Social emotion classification; recursive neural network; dependency embedding; topic model

ACM Reference Format:
Chang Wang, Bang Wang, Wei Xiang and Minghua Xu. 2019. Encoding Syntactic Dependency and Topical Information for Social Emotion Classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331287

*Chang Wang, Bang Wang and Wei Xiang are with the School of Electronic, Information and Communications, HUST; Minghua Xu is with the School of Journalism and Information Communication, HUST. This work is supported in part by National Natural Science Foundation of China (Grant No. 61771209)

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331287

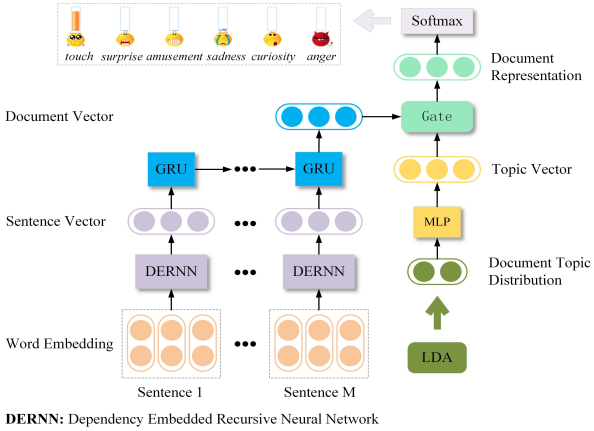

Figure 1: The framework of our proposed model.

1 INTRODUCTION

Some early methods have been proposed for social emotion classification, including the word-emotion models [2], which discover the direct relations between words and emotions based on the features of individual words. However, those word-emotion models cannot distinguish different emotions of a same word in different contexts. Some topic-emotion models [1, 6] utilize topic models to discover topical information of a document and model the relations between topics and emotions. However, they treat each word separately, yet ignoring some inter-word information like semantics.

Recently, some neural network-based models have been proposed for social emotion classification [4]. These models can automatically learn text features and generate a semantical document representation based on the *convolutional neural network* (CNN) [4] or *recurrent neural network* (RNN) [9]. But they have not fully utilized the syntactic feature of a sentence and the topical information of a document.

We argue that the syntactic dependency relations between words in a sentence are important for social emotion classification. Furthermore, we also support the claim in the existing work that the document topical information would help to distinguish the emotion of a same word in different contexts. So we design a new neural network model to include the both in document representation for social emotion classification.

Specifically, our proposed model includes a document encoding component and a topic encoding component. Fig. 1 presents the overall network framework. For document encoding, we design a *dependency embedded recursive neural network* (DERNN) to encode each syntactic dependency

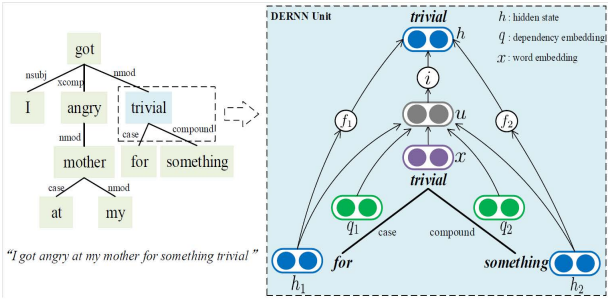

Figure 2: The left part is an example of dependency tree. The right part is the internal structure of the DERNN unit.

relation in between words of a sentence into the sentence vector in the lower layer, and we use a *gated recurrent unit* (GRU) to encode a document vector in the upper layer. For topic encoding, we apply a *multi-layer perceptron* (MLP) to transform the document topical distribution obtained from the *Latent Dirichlet Allocation* (LDA) model into a topic vector. Afterwards, to let the network adaptively decide the importance of the document vector and the topic vector, we propose to use a gate layer to obtain the final document representation from them yet with a gate controlling mechanism. Finally, we feed the document representation into a softmax layer to generate the predicted social emotion distribution. Experiment results on two public datasets have validated the superiority of our proposed model over the state-of-the-art ones in terms of higher average Pearson correlation coefficient and MicroF1 performance.

2 THE PROPOSED METHOD

1) Document Encoding: Since a document consists of one or more sentences, we first encode the syntactic dependency information in one sentence into the sentence vector in the lower layer network and then compose the document vector from sentence vectors in the upper layer network. We design a dependency embedded recursive neural network (DERNN) to model a sentence. It learns an embedding representation for each dependency relation (called dependency embedding) and encodes the syntactic dependency information into the sentence vector.

Before we input words into the network, we adopt a pre-trained word2vec model to transform each word into a word embedding, which is a low dimensional dense vector with real values. To construct the DERNN network, we use a dependency analysis tool for each sentence to obtain its dependency tree, where each node represents a word and each edge represents a dependency relation. The left part in Fig. 2 presents an example of dependency tree. Each parent node is connected with one or more child nodes and the dependency relations between them are marked above the edges. We model each node as a DERNN unit and its internal structure is illustrated in the right part of Fig. 2. We can see that the child hidden states h_1 and h_2, the parent word embedding x and the dependency embeddings q_1 and q_2 are input into the parent DERNN unit whose output is the hidden state h of the parent node. The modeling process starts from the bottom leaf nodes till the root node, and we treat the hidden state of the root node as the sentence vector.

Let w_n denote the nth word in a sentence and $\mathbf{x}_n \in \mathbb{R}^{d_w}$ is its word embedding, $C(w_n)$ is the child node set of w_n. The transition functions of the DERNN are as follows.

$$\tilde{\mathbf{h}}_n = \sum_{j \in C(w_n)} \mathbf{h}_j \quad (1)$$

$$\tilde{\mathbf{q}}_n = \sum_{j \in C(w_n)} \mathbf{q}_j \quad (2)$$

$$\mathbf{f}_j = \sigma(\mathbf{W}^{(f)}\mathbf{x}_n + \mathbf{U}^{(f)}\mathbf{h}_j + \mathbf{D}^{(f)}\mathbf{q}_j + \mathbf{b}^{(f)}) \quad (3)$$

$$\mathbf{i}_n = \sigma(\mathbf{W}^{(i)}\mathbf{x}_n + \mathbf{U}^{(i)}\tilde{\mathbf{h}}_n + \mathbf{D}^{(i)}\tilde{\mathbf{q}}_n + \mathbf{b}^{(i)}) \quad (4)$$

$$\mathbf{u}_n = tanh(\mathbf{W}^{(u)}\mathbf{x}_n + \mathbf{U}^{(u)}\tilde{\mathbf{h}}_n + \mathbf{D}^{(u)}\tilde{\mathbf{q}}_n + \mathbf{b}^{(u)}) \quad (5)$$

$$\mathbf{h}_n = tanh(\mathbf{i}_n \odot \mathbf{u}_n + \sum_{j \in C(w_n)} \mathbf{f}_j \odot \mathbf{h}_j) \quad (6)$$

where $j \in C(w_n)$ in Eq. (3), $\mathbf{W}^{(f,i,u)}, \mathbf{U}^{(f,i,u)}, \mathbf{D}^{(f,i,u)}, \mathbf{b}^{(f,i,u)}$ are learnable parameters and $\mathbf{h}_n \in \mathbb{R}^{d_h}$ is the hidden state of w_n. $\mathbf{q}_j \in \mathbb{R}^{d_q}$ is the dependency embedding of the dependency relation between the jth child node and the parent node. Because leaf nodes have no child nodes, they share an additional dependency embedding. All dependency embeddings are xavier uniform initialized and updated during the network training. The input gate $\mathbf{i}_n$ and the forget gate $\mathbf{f}_j$ are both dependent on the dependency embeddings. This allows the DERNN to forget the child words with unimportant dependency relations (like the punctuation relation) and remember the words with important relations (like the subject-predicate relation), to compose a sentence vector.

Regarding a document as a sequence of sentences, we use a gated recurrent unit to obtain a document vector from sentence vectors. The GRU transition functions are as follows.

$$\mathbf{z}_m = \sigma(\mathbf{W}^{(z)}\mathbf{s}_m + \mathbf{U}^{(z)}\mathbf{h}_{m-1} + \mathbf{b}^{(z)}) \quad (7)$$

$$\mathbf{r}_m = \sigma(\mathbf{W}^{(r)}\mathbf{s}_m + \mathbf{U}^{(r)}\mathbf{h}_{m-1} + \mathbf{b}^{(r)}) \quad (8)$$

$$\mathbf{u}_m = tanh(\mathbf{W}^{(u')}\mathbf{s}_m + \mathbf{U}^{(u')}(\mathbf{r}_m \odot \mathbf{h}_{m-1}) + \mathbf{b}^{(u')}) \quad (9)$$

$$\mathbf{h}_m = (1 - \mathbf{z}_m) \odot \mathbf{h}_{m-1} + \mathbf{z}_m \odot \mathbf{u}_m \quad (10)$$

where $\mathbf{s}_m$ is the sentence vector of the mth sentence and $\mathbf{h}_m \in \mathbb{R}^{d_{h'}}$ is its hidden state, and $d_h = d_{h'}$ in our work. The hidden state of the last sentence is viewed as the document vector $\mathbf{d}$.

2) Topic Encoding: Considering that social emotion is often related to the document topic, we propose to encode the document topical information into the document representation based on the LDA model, which regards a document as a mixture over various latent topics. After training a LDA model, we obtain the topic probability distribution of each document, denoted as $\mathbf{p} = \{pr(k)\}_{k=1}^{K}$, where K is the number of topics and k the topic index. $\mathbf{p}$ is treated as the original topic representation and then fed into a multi-layer perceptron followed by a *tanh* activation layer to get the topic vector $\mathbf{t}$, formulated as follows:

$$\mathbf{t} = tanh(\mathbf{W}^{(p)}\mathbf{p} + \mathbf{b}^{(p)}) \quad (11)$$

where $\mathbf{W}^{p} \in \mathbb{R}^{d_t \times K}$, d_t is the dimension of the topic vector. Notice that in our work, $d_t = d_h$. The MLP layer is used to discover the topic feature in the original topic distribution $\mathbf{p}$.

3) Gate Layer: After obtaining the document vector $\mathbf{d}$ and the topic vector $\mathbf{t}$, a straightforward strategy is to sum them to get the final document representation. But we argue that the syntactic dependency and topical information are not always the same important for social emotion classification. So we design a gate layer which has a document gate and a topic gate to combine the two vectors. The transition functions of the gate layer are computed as follows.

$$\mathbf{g_d} = \sigma(\mathbf{W}^{(g_d)}\mathbf{d} + \mathbf{U}^{(g_d)}\mathbf{t} + \mathbf{b}^{(g_d)}) \quad (12)$$

$$\mathbf{g_t} = \sigma(\mathbf{W}^{(g_t)}\mathbf{d} + \mathbf{U}^{(g_t)}\mathbf{t} + \mathbf{b}^{(g_t)}) \quad (13)$$

$$\mathbf{v} = tanh(\mathbf{g_d} \odot \mathbf{d} + \mathbf{g_t} \odot \mathbf{t}) \quad (14)$$

where $\mathbf{g_d}$ is the document gate, $\mathbf{g_t}$ is the topic gate and $\mathbf{W}^{(g_d,g_t)}, \mathbf{U}^{(g_d,g_t)}, \mathbf{b}^{(g_d,g_t)}$ are the learnable parameters. $\odot$ represents element-wise multiplication. The gate layer allows the network to adaptively assign different importance to the document vector and the topic vector, when composing the final document representation $\mathbf{v} \in \mathbb{R}^{d_v}, d_v = d_t = d_h$.

4) Social Emotion Classification: In this part, we first add a linear layer to transform the document representation $\mathbf{v}$ into a label vector whose dimension is the number of the emotion labels E. Then the label vector is fed into a softmax layer to get the predicted probability vector $\hat{\mathbf{y}}$.

$$\hat{\mathbf{y}} = softmax(\mathbf{W}^{(l)}\mathbf{v} + \mathbf{b}^{(l)}) \quad (15)$$

where $\mathbf{W}^l \in \mathbb{R}^{E \times d_v}$ and $\mathbf{b}^l \in \mathbb{R}^E$ are the parameters in the linear layer.

Considering the evaluation objective in social emotion classification is an emotion probability distribution, other than a single most likely emotion label. So for the network training, we use the Kullback-Leibler divergence between the gold emotion distribution $\mathbf{y}$ and the predicted emotion distribution $\hat{\mathbf{y}}$ as the loss function:

$$Loss(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{E}\sum_{c=1}^{E} y_c(\log(y_c) - \log(\hat{y}_c)) \quad (16)$$

Finally, we train the whole network with the Adam optimizer.

3 EXPERIMENT

Datasets: We experiment on two public datasets: SinaNews [5] and ISEAR [7]. SinaNews is a Chinese dataset which contains 5,258 hot news collected from the social channel of the news website (www.sina.com). To be consistent with the baseline methods [5], we use 3,109 articles as the training set and 2,149 articles as the testing set. ISEAR is an English dataset which consists of 7,666 samples. Each sample is a paragraph of text tagged by an emotion label. Like [3], we randomly select 60% of the samples as the training set and the remaining 40% as the testing set.

Settings: For the Chinese dataset SinaNews, we train a 100-dimensional word2vec model with the Chinese Wikipedia corpus[1]. The LTP toolkit[2] provided by HIT is used for dependency parsing. There are in total 15 dependency relations predefined in the LTP, and we define one dependency embedding for each relation. For the English dataset ISEAR, we directly use the 300-dimensional English word2vec model provided by Google[3]. Stanford Parser[4] is used to construct dependency trees. Since there are too many kinds of dependency relations in the parsing result of Stanford Parser, to balance the occurrence number of each relation, we map all relations into 9 categories according to universal dependencies v1[5]. Other experiment parameters are setting as follows: For SinaNews and ISEAR, the topic number K in LDA is 30 and 10, $d_h = d_q = d_t$ are 200 and 100, and the learning rate is both 0.001, the batch size is both 20.

Evaluation Metrics: Considering label distributions are very imbalanced in the datasets, we adopt the Micro-averaged F1 score ($MicroF1$) to reflect the accuracy of the predicted top-ranked emotion label [3]. And the average Pearson correlation coefficient (AP) is used to measure the divergence between the predicted emotion probability distribution and the ground truth distribution [3].

$$MicroF1 = \frac{\sum_{d \in \mathcal{D}_{test}} \mathbb{I}_d}{|\mathcal{D}_{test}|}, \quad \mathbb{I}_d = \begin{cases} 1, & \text{if } \hat{e}^d_{top} = e^d_{top}, \\ 0, & \text{otherwise,} \end{cases} \quad (17)$$

where e^d_{top} is the actual top-ranked emotion triggered by an article d and $\hat{e}^d_{top}$ is the predicted one. $\mathcal{D}_{test}$ denotes the testing set. AP between the predicted emotion distribution $\hat{\mathbf{y}}$ and the ground truth distribution $\mathbf{y}$:

$$AP = \frac{\sum_{\mathcal{D}_{test}} r(\hat{\mathbf{y}}, \mathbf{y})}{|\mathcal{D}_{test}|}, \quad r(\hat{\mathbf{y}}, \mathbf{y}) = \frac{cov(\hat{\mathbf{y}}, \mathbf{y})}{\sqrt{var(\hat{\mathbf{y}})var(\mathbf{y})}}. \quad (18)$$

where r denotes the Pearson correlation coefficient and cov denotes the covariance operation.

Comparison models: We compare our proposed model, Gated DR-G-T (Gated DERNN-GRU-Topic), with the following models. The first group contains the baseline models from the literature, including the SWAT [2], Emotion Topic Model (ETM) [1], Contextual Sentiment Topic Model (CSTM) [6], Social Opinion Mining model (SOM) [5], 1-HNN-BTM [3], and CNN and CNN-SVM [5].

The second group is the LSTM models implemented by us.Hierarchical LSTM (H-LSTM) is a hierarchical structure of two LSTM networks which are used for sentence modeling and document modeling, respectively. Child-Sum Tree-LSTM - LSTM (T-LSTM) uses the Child-Sum Tree-LSTM (a tree-structured LSTM proposed by [8]) to model sentences and the LSTM to model the document.

The third group is the variants of Gated DR-G-T for model analysis. The RNN-GRU (R-G) uses a recursive neural network in the lower layer, which removes dependency embeddings compared to DERNN. The GRU is used to model the document. The DERNN-GRU (DR-G) uses the DERNN proposed in this work to model sentences and the GRU to model the document. Compared to Gated DR-G-T, the DERNN-GRU-Topic (DR-G-T) does not use the gate layer.

Experimental Results: Table 1 presents the experiment results, where our proposed Gated DR-G-T model outperforms all the other models. Specifically, compared to the best

[1] https://dumps.wikimedia.org/
[2] https://www.ltp-cloud.com/
[3] https://code.google.com/archive/p/word2vec/
[4] https://nlp.stanford.edu/software/lex-parser.shtml
[5] http://universaldependencies.org/docsv1/u/dep/all.html

Table 1: Experiment results on the two datasets.

Models	SinaNews		ISEAR	
	MicroF1	AP	MicroF1	AP
SWAT [2]	38.97	0.40	26.29	0.21
ETM [1]	54.19	0.49	48.79	0.35
CSTM [6]	40.74	0.43	28.23	0.19
1-HNN-BTM [3]	-	-	51.21	0.40
CNN [5]	51.23	-	-	-
CNN-SVM [5]	52.63	-	-	-
SOM [5]	58.59	0.64	-	-
H-LSTM	61.69	0.68	59.13	0.56
T-LSTM	62.57	0.68	59.98	0.57
R-G	63.55	0.69	60.04	0.57
DR-G	64.48	0.70	59.55	0.56
DR-G-T	64.39	0.70	60.06	0.57
Gated DR-G-T	65.20	0.71	60.44	0.57

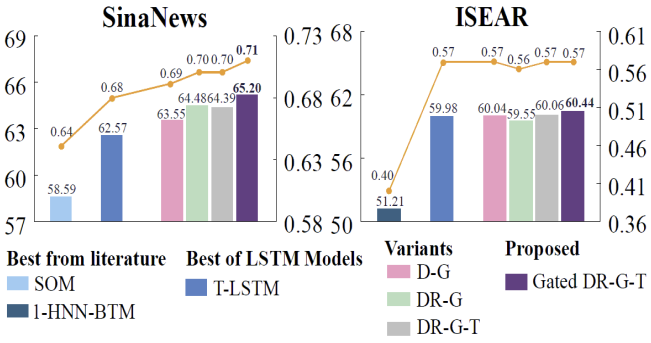

Figure 3: Comparison between the proposed Gated DR-G-T, its variants, the best LSTM Model and the best model from the literature. The left vertical axis and the bar stand for $MicroF1$; The right vertical axis and the line stand for AP.

model from the literature (cf., Fig. 3), Gated DR-G-T improves MicroF1 by 6.61% and AP by 0.07 on SinaNews, and improves MicroF1 by 9.23% and AP by 0.17 on ISEAR. We attribute the improvements to the leverage of encoding the syntactic dependency and topical information into document representation. We note that the performance of the two LSTM models performs better than the baseline models from the literature. A possible reason is that they adopt a hierarchical structure and can learn the word-level and sentence-level semantic features respectively. Although the T-LSTM model considers the syntactic dependency structure of a sentence, it ignores the specific dependency relation in between words. On the one hand, the DERNN in our model can encode each specific dependency relation into sentence vectors. On the other hand, our model also values the topical information via LDA-based topic encoding. Furthermore, with the help of the gate layer, the final document representation can well balance the syntactic dependency and topical information for social emotion classification.

Fig. 3 compares the results between the Gated DR-G-T and its variants. We first observe that DR-G performs better than R-G on SinaNews, where the latter does not concern the specific dependency relations. This indicates that encoding syntactic dependency contributes to the performance improvement. But the result of DR-G is worse than R-G in ISEAR. A possible reason is that there are too many short texts in ISEAR and the number of sentences is not enough to well train the dependency embedding. Another observation is that although DR-G-T uses the topical information, it does not show an obvious advantage over DR-G on the two datasets. Although the dependency feature and topical information enrich the semantics of a document representation, they should be carefully integrated. The straightforward summation might add some noises into the final document representation. Since the proposed Gated DR-G-T applies a gate layer to control their weights in the final document representation, the results reveal that it can perform better than the naive DR-G-T.

4 CONCLUSION

In this paper, we have proposed a new neural network model for social emotion classification. For document encoding, the DERNN encodes syntactic dependency relations into sentence vectors and the GRU transforms sentence vectors into a document vector. For topic encoding, the MLP transforms the document topic distribution into a topic vector. Finally, a gate layer is used to compose the final document representation from the gated summation of document vector and topic vector. Experiments on two public datasets show that compared with the state-of-the-art models, the proposed model can improve the classification performance in terms of higher MicroF1 and AP on two public datasets.

REFERENCES

[1] S. Bao, S. Xu, L. Zhang, R. Yan, Z. Su, D. Han, and Y. Yu, "Mining social emotions from affective text," *IEEE transactions on knowledge and data engineering*, vol. 24, no. 9, pp. 1658–1670, 2012.

[2] P. Katz, M. Singleton, and R. Wicentowski, "Swat-mp: the semeval-2007 systems for task 5 and task 14," in *Proceedings of the 4th international workshop on semantic evaluations*. Association for Computational Linguistics, 2007, pp. 308–313.

[3] X. Li, Y. Rao, H. Xie, R. Y. K. Lau, J. Yin, and F. L. Wang, "Bootstrapping social emotion classification with semantically rich hybrid neural networks," *IEEE Transactions on Affective Computing*, vol. 8, no. 4, pp. 428–442, 2017.

[4] X. Li, Y. Rao, H. Xie, X. Liu, T.-L. Wong, and F. L. Wang, "Social emotion classification based on noise-aware training," *Data & Knowledge Engineering*, 2017.

[5] X. Li, Q. Peng, Z. Sun, L. Chai, and Y. Wang, "Predicting social emotions from readers perspective," *IEEE Transactions on Affective Computing*, no. 1, pp. 1–1, 2017.

[6] Y. Rao, "Contextual sentiment topic model for adaptive social emotion classification," *IEEE Intelligent Systems*, no. 1, pp. 41–47, 2016.

[7] K. R. Scherer and H. G. Wallbott, "Evidence for universality and cultural variation of differential emotion response patterning." *Journal of personality and social psychology*, vol. 66, no. 2, p. 310, 1994.

[8] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[9] X. Zhao, C. Wang, Z. Yang, Y. Zhang, and X. Yuan, "Online news emotion prediction with bidirectional lstm," in *International Conference on Web-Age Information Management*. Springer, 2016, pp. 238–250.