Topic-Enhanced Capsule Network for Multi-Label Emotion Classification

Hao Fei, Donghong Ji, Yue Zhang, and Yafeng Ren

***Abstract*—Identifying multiple emotions in a piece of text is an important research topic in the NLP community.Existing methods usually model the task as a multi-label classification problem, while these work has two issues. First, these methods fail to leverage the topic information of the text, which has been shown to be effective for sentiment analysis task. Second, different parts of the text can contribute differently to predicting different emotion labels, so the proposed model needs to capture effective features for each corresponding emotion, which is not considered by existing models. To tackle these problems, we propose a topic-enhanced capsule network, which contains two main parts: a variational autoencoder and a capsule module, for multi-label emotion detection task. Specifically, the variational autoencoder can learn the latent topic information of the text, and the capsule module can capture rich features for corresponding emotion. Experimental results on two benchmark datasets show that the proposed model achieves the current best performance, outperforming previous methods and strong baselines by a large margin.**

***Index Terms*—Information extraction, emotion detection, neural networks, topic model, sentiment analysis.**

I. Introduction

AUTOMATIC emotion detection is one important task in natural language processing (NLP) [1]–[5], which facilitates a wide range of downstream applications such as chatbots [6], [7], stock prediction [8], [9] and policy studies [10], [11], etc. In social media, people tend to express multiple emotions in one piece of text. As shown in Table I, multiple emotions co-exist in sentences such as S1:"I will watch the horror movie Blair Witch." Specifically, there are three emotions, including **anticipation**, **joy** and **fear** expressed in S1.

TABLE I
Examples of Multiple Emotions in a Sentence

Training examples:		
S1	[**anticipation, joy, fear**]	I will watch the horror movie **Blair Witch**.
S2	[**surprise, fear, sadness, optimism**]	**Halloween** movies frightened me much. Now nightmare still on...
S3	[**anticipation, love, pessimism, disgust**]	How's the new **Batman Telltale** Series? Looks good but I'm growing weary of the gaming style.
Test examples:		
S4	[**fear, joy, love**]	Love the new movie **Blair Witch**.
S5	[**anticipation, fear, joy, optimism, trust**]	**Halloween** party coming soon! But I'll learn to overcome myself.

Similarly, S2 contains four emotions: **surprise**, **fear**, **sadness** and **optimism**. Identifying multiple co-existing emotions in a sentence remains a challenging task.

Existing methods are mainly divided into two classes: methods for multi-label classification and emotion ranking methods. The former regards the task as a general multi-label classification problem, and existing models mainly use Binary Relevance [12], Classier Chains [13], ML-KNN [12] or a Joint Binary Classifier [14]–[16]. The emotion ranking methods directly learn the emotion distribution and the relevant emotion ranking [17], [18].

Different from the existing methods, we have two new observations for multi-label emotion detection. First, contextual information, or prior information, such as topic information retained in different sentences, can be leveraged to improve the performance of the task. Taking S4 of Table I for example, a model may fail to capture the emotion **fear** because there are no clues explicitly indicating the emotion. It can be observed that the representation of the entity *Blair Witch* in sentence S1 can be strengthened from its co-current word horror movie under one common topic, which is closely related to the emotion **fear**. If we can first capture such information, the model can easily infer the emotion **fear** for S4 based on the word **Blair Witch**. Similarly, this observation holds for sentence S5 and S2.

Second, we find that there is a relatively large number of emotions, which can be independent, and the correlation between which and the input text can be complex. Specifically, useful clue words for each emotion can be scattered, or mixed in a sentence, and one word may support multiple emotion labels. Taking S3 in Table 1 as example, the clues *Batman Telltale*, *but* and *weary* indicating the emotion **pessimism** are scattered, and also surrounding the cue *good* that indicates the emotion **love**. The word *weary* entails two emotions, **pessimism** and **disgust**.

Manuscript received December 4, 2019; revised April 12, 2020 and May 22, 2020; accepted June 5, 2020. Date of publication June 10, 2020; date of current version June 23, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61702121 and 61772378, in part by the National Philosophy Social Science Major Bidding Project under Grant 11&zd189, in part by the Research Foundation of Ministry of Education of China under Grant 18JZD015, in part by the Key Project of State Language Commission of China under Grant ZDI135-112, in part by the Guangdong Basic and Applied Basic Research Foundation of China under Grant 2020A151501705, and in part by the Bidding Project of GDUFS Laboratory of Language Engineering and Computing under Grant LEC2018ZBKT004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong Cui. *(Corresponding author: Yafeng Ren.)*

Hao Fei and Donghong Ji are with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: hao.fei@whu.edu.cn; jidonghong@whu.edu.cn).

Yue Zhang is with the School of Engineering, Westlake University, Hangzhou 310024, China (e-mail: yue.zhang@wias.org.cn).

Yafeng Ren is with the Laboratory of Language Engineering and Computing, Guangdong University of Foreign Studies, Guangzhou 510420, China (e-mail: renyafeng@whu.edu.cn).

Digital Object Identifier 10.1109/TASLP.2020.3001390

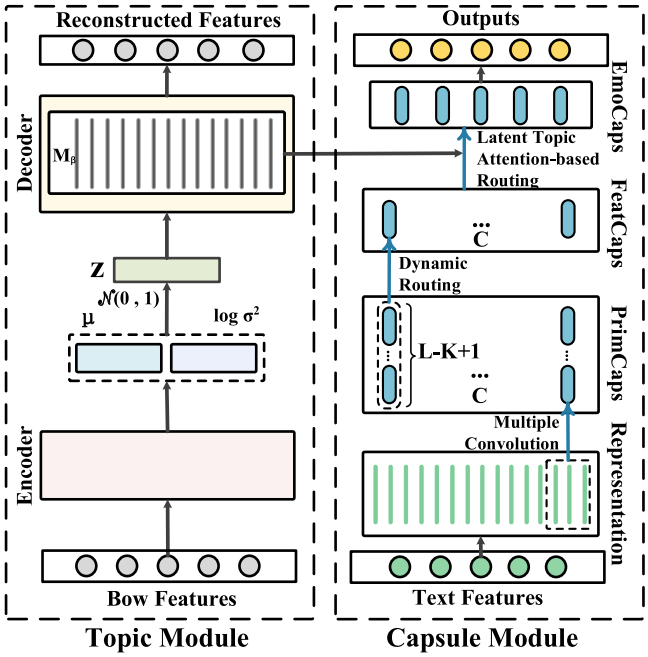


Fig. 1. The overall framework of the proposed model.

This observation suggests that a model should have strong ability to extract effective features for each corresponding emotion in a sentence. However, existing methods typically employ a simple network such as recurrent neural network (RNN) or convolutional neural network (CNN) to encode sentences and extract features for all emotions simultaneously.

Although probabilistic topic models such as LSA [19] and LDA [20] have been widely used for topic modelling, they cannot be directly integrated into neural networks with a joint manner. In contrast, variational autoencoder (VAE) has been shown to have strong ability for latent semantic learning [21], [22]. Inspired by previous work [23], we exploit VAE to model the topics as latent variables $\boldsymbol{Z}$. In addition, we employ the capsule network to learn rich features for each corresponding emotion. The reasons are two-fold. First, unlike RNN or CNN, the capsule network embeds features into capsule vectors, where each scalar element describes a specific property of a feature. Such encapsulated features can benefit the learning of the part-whole relationship for the separate emotion labels [24]. Second, the routing algorithm can help iteratively integrate the features [25], capturing the most relevant contexts for representing each emotion.

In this paper, we propose a Topic-enhanced Capsule (TECap) network for multi-label emotion classification. As shown in Figure 1, TECap consists of two main components: a topic module and a capsule module. The former learns the latent topics and keywords by reconstructing the bag-of-word (BoW) input via variational autoencoder (VAE) [21]. The latter captures encapsulated features separately for each emotion from the low level to the high level via three deep capsule layers. During the semantic feature transferring between the last two capsule layers, topic information is utilized by a latent topic attention-based routing algorithm, in order to learn the task-relevant features for corresponding emotions. Finally, the capsule module computes the probability for each emotion label independently. Note that the two components can be trained jointly in an end-to-end learning, so that TECap is able to learn latent topic information without external knowledge, facilitating multi-label emotion classification.

We conduct extensive experiments on the SemEval 2018 task 1 C English dataset and the Ren-CECps Chinese dataset. Results show that our proposed model significantly outperforms strong baselines, achieving the state-of-the-art performance. The remainder of the paper is structured as follows. In Section II, we review related work about multi-label emotion classification, variational models and capsule networks. Section III presents our TECap model in detail. In Section IV, we describe details about the experimental settings and experimental results. Finally, we give our conclusion in Section V.

II. Related Work

A. Multi-Label Emotion Classification

Existing methods can be divided into two classes: methods for general multi-label classification and emotion ranking methods. The former regards the task as a common multi-label classification problem, and uses various types of models such as Binary Relevance [12], Classier Chains [13], ML-KNN [12] and Joint Binary Classifier [14] etc., for making predictions. For example, Wang *et al.* (2016) propose the TDNN framework by constructing a convolutional neural network for multi-class classification [26]. Yu *et al.* (2018) propose a transfer learning architecture to improve the performance of multi-label emotion classification. However, these methods fail to leverage contextual information and prior information such as topic information, which can be useful for improving the performance of the task [27]–[29].

Different from the above methods, the emotion ranking methods try to directly learn the emotion distribution and relevant emotion ranking [17], [18], [30]. For instance, Zhou *et al.* (2016) propose an emotion distribution learning method, which first learns the relations between emotions, and then conducts multi-label emotion classification by incorporating these label relations into the cost function [17]. Zhou *et al.* (2018) explore a relevant emotion ranking model by re-designing the objective function [18]. Yang *et al.* (2018) present a model for relevant emotion ranking by transferring knowledge from topic models to the target task of multi-label emotion [30]. However, these models fail to learn effective features for each corresponding emotion, which is crucial for correctly predicting all emotions.

B. Variational Models

Our proposed method is also related to variational models in NLP applications [22], [23], [31]. In recent years, variational and its variant models have been widely utilized for various tasks [32]–[35]. For example, Bowman *et al.* (2015) introduce a RNN-based VAE model for generating diverse and coherent sentences. Miao *et al.* (2016) propose a neural variational framework by incorporating multilayer perceptrons (MLP), CNN and

RNN for generative models of the text. Bahuleyan *et al.* (2017) design an attention-based variational seq2seq model. More recently, Zhang *et al.* (2019) exploit VAE to integrate sentence with syntactic trees for improving the grammar of generated sentences. Different from these model for the generation of texts, in this study, we employ a VAE model to reconstruct the input, during which we learn the latent topic information for facilitating downstream prediction.

C. Capsule Networks

Capsule networks were first proposed to better learn the part-whole relationship [36]. Sabour *et al.* (2017) introduce a dynamic routing algorithm for capsule networks to replace the pooling operation with routing-by-agreement [37], and they leverage a capsule network for image classification. Due to its characteristics of encapsulated features, recently, many efforts are paid for exploiting capsule networks for various NLP tasks [38]–[43]. For example, Yang *et al.* (2018) utilize the capsule network for text classification [38]. Wang *et al.* (2018) use a RNN-based capsule network for sentiment analysis [44]. Zhang *et al.* (2018) present an attention RNN with capsule network for relation classification [45]. Chen *et al.* (2019) employ the capsule architecture to retrieve rich features for sentiment analysis [24]. More recently, Chen *et al.* (2019) use capsule networks to enhance the interactions between arguments and predicates for semantic role labeling tasks [25]. To our knowledge, we are the first to investigate multi-label emotion classification via a capsule network.

III. Method

As shown in Figure 1, the proposed model consists of two components: a topic module and a capsule module. These two modules are linked via the usage of topic representations. The topic module first learns latent topics and keywords by the reconstruction of the BoW input via VAE. Then, the capsule module captures encapsulated features for each emotion from low level to high level via three deep capsule layers. During the semantic feature transferring between the last two capsule layers, the topic information is utilized by the latent topic attention-based routing algorithm to capture the most relevant features for each corresponding emotion. Finally, the capsule module predicts the probability for each emotion label independently.

A. Topic Module

1) Encoder: The input of the topic module is the bag-of-word (BoW) features of a sentence. Given a sentence $\boldsymbol{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_L\}$ (L represents the sentence length), each token $\boldsymbol{s}$ is processed into a BoW representation $\boldsymbol{x}_{BoW} \in \mathbb{R}^V$ (V is the vocabulary size). Specifically, the encoder $f_e(\cdot)$ consists of multiple non-linear hidden layers, transforming $\boldsymbol{x}_{BoW}$ into prior parameters μ and σ:

$$\begin{aligned} \boldsymbol{\mu} &= f_e^{\mu}(\boldsymbol{x}_{BoW}), \\ log\boldsymbol{\sigma} &= f_e^{\sigma}(\boldsymbol{x}_{BoW}). \end{aligned} \tag{1}$$

We define the latent variable $\boldsymbol{Z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \epsilon$, where ϵ is Gaussian noise variable sampled from $\mathcal{N}(0, 1)$ [22]. The value of $\boldsymbol{Z} \in \mathbb{R}^T$ (T is the topic number) is then normalized by a softmax function, which reflects the distribution of T latent topics.

2) Decoder: We use variational inference [21]–[23] to approximate a posterior distribution over $\boldsymbol{Z}$. A feedforward neural network is used to reconstruct $\boldsymbol{Z}$ into $\widehat{\boldsymbol{x}}_{BoW}$:

$$\widehat{\boldsymbol{x}}_{BoW} = softmax(f_d(\boldsymbol{Z}; \boldsymbol{M})), \tag{2}$$

where $\boldsymbol{M} \in \mathbb{R}^{T \times V}$ is the kernel of $f_d(\cdot)$, and the embedding of a rich **topic-keyword** representation, which will later be leveraged into the capsule module.

$\boldsymbol{M}$ is normalized along each topic $t = (0, \ldots, T)$:

$$\boldsymbol{M}_{t,v} = \frac{\exp(\boldsymbol{M}_{t,v})}{\sum_{v'=0}^{V} \exp(\boldsymbol{M}_{t,v'})}, \tag{3}$$

where $\boldsymbol{M}_t$ represents the t-th topic by a keyword distribution over the vocabulary.

3) Learning: Following Le *et al.* (2018), we learn the parameters of VAE by maximizing the variational lower bound on the marginal log likelihood of features:

$$\begin{aligned} logp_\theta(\boldsymbol{x}_{BoW}) \geq\ & \mathbb{E}_{\boldsymbol{Z} \sim q_\phi(\boldsymbol{Z}|\boldsymbol{x}_{BoW})}[logp_\theta(\boldsymbol{x}_{BoW}|\boldsymbol{Z})] \\ & - KL(q_\phi(\boldsymbol{Z}|\boldsymbol{x}_{BoW})||p(\boldsymbol{Z})), \end{aligned} \tag{4}$$

where ϕ and θ are the parameters of the encoder and the decoder, respectively, and the KL-divergence term ensures that the distributions $q_\phi(\boldsymbol{Z}|\boldsymbol{x}_{BoW})$ is near to the prior probability $p(\boldsymbol{Z})$, which is a standard normal distribution $\mathcal{N}(0, 1)$.

Since the training objective of the decoder is to reconstruct the input, it has direct access to the source features. Thus, when the decoder is trained, we assume that $q(\boldsymbol{Z}|\boldsymbol{x}_{BoW}) = q(\boldsymbol{Z}) = p(\boldsymbol{Z})$, which means that the KL loss is zero. It makes the latent variables $\boldsymbol{Z}$ fail to capture useful information. The reason behind lies in the stronger decoder. To combat this, we employ KL cost annealing and word dropout for the encoder [46]. For instance, we randomly replace some of the input words (e.g., 5%) with a 'UNK' token to weaken the auto-regressive decoder and anneal the KL divergence term.

B. Capsule Module

The capsule module contains three capsule layers: primary capsule, feature capsule and emotion capsule. All of them can learn the features from the low level to the high level.

1) Input Representation: Given a sentence $\boldsymbol{S}$, each token $\boldsymbol{s}$ is transformed into a word vector $\boldsymbol{e} \in \mathbb{R}^{d_e}$ (d_e is embedding dimension) via a look-up table. Then, a BiLSTM is used to encode forward and backward hidden representations $\boldsymbol{h}_{f_i}$ and $\boldsymbol{h}_{b_i}$, respectively. Finally, we concatenate the two hidden states as a sequence representation $\boldsymbol{h} \in \mathbb{R}^{d_h \times L}$.

2) Primary Capsule Layer: We first define $p_i \in \mathbb{R}^{d_p}$ (d_p is the dimension of capsule) as the instantiated parameters of a capsule. The primary capsule (PrimCap) as the first capsule layer is built to extract the n-gram features. We use $\boldsymbol{W}_{pr} \in \mathbb{R}^{K \times d_h}$ to denote a filter for the convolution, where K is the n-gram width over a sentence. We perform the convolution with the filter $\boldsymbol{W}_{pr}$

to produce each primary feature map $f_i^1 \in \mathbb{R}$:

$$f_i^1 = f(\boldsymbol{h}_{i:i+K} \circ \boldsymbol{W}_{pr} + b), \tag{5}$$

where $f(\cdot)$ is the activate function and $\circ$ denotes the element-wise multiplication. We perform the convolution d_p times with the kernel $\boldsymbol{W}_{pr}$ to generate the primary feature capsules $\boldsymbol{P}^1 \in \mathbb{R}^{d_p \times (L-K+1)}$, using the squash function $g(\cdot)$:

$$\boldsymbol{P}_j^1 = g(\boldsymbol{f}^1), \tag{6}$$

$$g(\mathbf{x}) = \frac{\|x\|^2}{0.5 + \|x\|^2} \frac{x}{\|x\|}. \tag{7}$$

Then we repeat the convolution operation C times to obtain a total number of C channels of features. Here, the primary capsule is represented as $\boldsymbol{P}^1 \in \mathbb{R}^{d_p \times (L-K+1) \times C}$.

3) Feature Capsule Layer: We set the second feature capsule (FeatCap) layer to retrieve semantic features at higher level. The PrimCap $\boldsymbol{P}^1$ is connected to the FeatCap $\boldsymbol{P}^2$ via **Dynamic Routing** (**DR**) [37]. An intermediate vector $\overline{\boldsymbol{P}^{\boldsymbol{1}}_{\boldsymbol{j|i}}}$ is first computed by $\overline{\boldsymbol{P}^{\boldsymbol{1}}_{\boldsymbol{j|i}}} = \boldsymbol{W}_{\boldsymbol{ij}} \boldsymbol{P}^{\boldsymbol{1}}_{\boldsymbol{i}}$. The connection value is:

$$\boldsymbol{S}_j = \sum_i c_{ij} \overline{\boldsymbol{P}^{\boldsymbol{1}}_{\boldsymbol{j|i}}} \quad (i \in [1, \ldots, (L-K+1) * C]), \tag{8}$$

where the coupling coefficients c_{ij} are obtained iteratively from an originally initiated value b_{ij} via:

$$c_{ij} = \underset{j}{softmax}(b_{ij}) \quad (i \in [1, \ldots, C]). \tag{9}$$

We squash the capsule into a range of [0,1]:

$$\boldsymbol{P}^2 = g(\boldsymbol{S}_j). \tag{10}$$

To perform dynamic routing, we update the original values b_{ij} for c_{ij}. The calculations are as follows:

$$a_{ij} = \overline{\boldsymbol{P}^{\boldsymbol{1}}_{\boldsymbol{j|i}}} \cdot \boldsymbol{P}^2, \tag{11}$$

$$b_{ij} = b_{ij} + a_{ij}. \tag{12}$$

Overall, dynamic routing is executed iteratively along the equation chain [(9)→(8)→(10)→(11)→(12)], forming a non-linear mapping $\boldsymbol{P}^{\boldsymbol{1}}_{\boldsymbol{j|i}} \mapsto \boldsymbol{P}^2_j$. In the FeatCap layer, rich semantic features are captured among C categories of capsules.

4) Emotion Capsule Layer: The emotion capsule (EmoCap) layer is built for extracting the semantic features captured at the FeatCap layer that is the most relevant to each corresponding emotion. We achieve this by introducing a novel **Latent Topic Attention-based Routing** (**LTAR**) algorithm, during the semantic feature transferring.

Based on the **DR** algorithm, we leverage the latent topic-keyword representation learnt at the topic module into capsule transforming. Specifically, for each keyword representation $\boldsymbol{M}_t$ under the t-th topic, we compute the relatedness between $\boldsymbol{M}_t$ and the raw capsule representation $\overline{\boldsymbol{P}^{\boldsymbol{3}}_{\boldsymbol{j}}}$, and then maintain the weighted representation as a new capsule representation $\boldsymbol{P}^3_j$. The attention computation can be formulated as:

$$u_t = \boldsymbol{V}^T tanh(\boldsymbol{W}_{a_1} \overline{\boldsymbol{P}^{\boldsymbol{3}}_{\boldsymbol{j}}} + \boldsymbol{W}_{a_2} \boldsymbol{M}_t + b), \tag{13}$$

$$\boldsymbol{\alpha} = softmax(\boldsymbol{u}), \tag{14}$$

Algorithm 1: Latent Topic Attention-based Routing.

Input: FeatCaps $\boldsymbol{P}^2$, topic-keyword representation $\boldsymbol{M}$, routing iteration r.
Output: EmoCaps $\boldsymbol{P}^3$

$b_{ij} = 0$.
$\overline{\boldsymbol{P}^{\boldsymbol{2}}_{\boldsymbol{j|i}}} = \boldsymbol{W}_{\boldsymbol{ij}} \boldsymbol{P}^{\boldsymbol{2}}_{\boldsymbol{i}}$.
for each iteration in r **do**
  $c_i = softmax(b_i)$.
  $\boldsymbol{S}_j = \sum_i c_{ij} \overline{\boldsymbol{P}^{\boldsymbol{2}}_{\boldsymbol{j|i}}}$.
  $\overline{\boldsymbol{P}^{\boldsymbol{3}}_{\boldsymbol{j}}} = g(\boldsymbol{S}_j)$.
  for each topic t in T **do**
    $u_t = \boldsymbol{V}^T tanh(\boldsymbol{W}_{a_1} \overline{\boldsymbol{P}^{\boldsymbol{3}}_{\boldsymbol{j}}} + \boldsymbol{W}_{a_2} \boldsymbol{M}_t + b)$.
  end for
  $\boldsymbol{\alpha} = softmax(\boldsymbol{u})$.
  $\boldsymbol{P}^3_j = \sum_{t=0}^{T} \alpha_t \boldsymbol{M}_t$.
  $a_{ij} = \overline{\boldsymbol{P}^{\boldsymbol{2}}_{\boldsymbol{j|i}}} \cdot \boldsymbol{P}^3$.
  $b_{ij} = b_{ij} + a_{ij}$.
end for

$$\boldsymbol{P}^3_j = \sum_{t=0}^{T} \alpha_t \boldsymbol{M}_t, \tag{15}$$

where $\boldsymbol{V}^T, \boldsymbol{W}_{a_1}$ and $\boldsymbol{W}_{a_2}$ are model parameters. During the routing process, we iteratively amend and optimize the connecting strengths between FeatCaps and EmoCaps. By enriching the sentence with such extended contexts, the capsule vectors can be more related to the corresponding emotions.

Similarly, we map the FeatCap layer into the EmoCap layer via $\boldsymbol{P}^{\boldsymbol{2}}_{\boldsymbol{j|i}} \mapsto \boldsymbol{P}^3_j$, where $i = [1, \ldots, C]$, $j = [1, \ldots, E]$, and E is the number of total emotion labels in the task. The routing process of **LTAR** is demonstrated in Algorithm 1.

C. Prediction and Training

In EmoCap layer, the length of each capsule vector represents the probability of each emotion label. We use a separate margin loss for learning the j-th emotion capsule:

$$\begin{aligned} L_j &= Y_j max(0, (B + \gamma^+) - \|\boldsymbol{P}^3_j\|)^2 + \\ &= \lambda(1 - Y_j) max(0, \|\boldsymbol{P}^3_j\| - (B - \gamma^-))^2, \end{aligned} \tag{16}$$

where $Y_j = 1$ if the j-th gold emotion label is presented. $\gamma^+ = 0.9$ and $\gamma^- = 0.1$ define the top and bottom margins, respectively. $\lambda = 0.5$ is used for the absent emotion labels and $B = 0.5$ indicates the threshold of the margin. In the test phase, emotion label is assigned only if the corresponding probability is larger than the threshold B. The total loss of the capsule module is $L_{all} = \sum_{j=1}^{E} L_j$.

Note that the topic module and the capsule module can be jointly trained. However, directly training the whole framework with cold-start can be difficult and cause high variance. Thus we first pre-train the topic module until it is close to the convergence via Eq.4. Afterwards, we jointly train all the components via Eq.4 and Eq.16. Once the classification loss is close to the convergence, we again train the topic module alone, until it

TABLE II
STATISTICS OF THE DATASETS. **# NCO.** REPRESENTS THE NUMBER OF CO-EXISTING EMOTION LABELS IN A SENTENCE

Dataset	3 co.(%)	2 co.(%)	1 co.(%)	Train:Dev:Test
RCECps	1,824(5.2)	11,416(32.5)	18,812(53.6)	24,567:3,510:7,019
Sem18	3,419(31.1)	4,442(40.4)	1,563(14.2)	6,838:886:3,259

converges. We then train the overall model. We keep such training strategy until it reaches plateau [47].

IV. EXPERIMENTS

A. Datasets

We conduct experiments on two benchmark datasets, including the English dataset SemEval 2018 (Sem18) [48] and the Chinese dataset Ren-CECps (RCECps) [49]. Table II shows statistics of the datasets. In Sem18, there are totally 35,096 documents, which are collected from Twitter. Each document is annotated with 11 basic emotion labels from writer's perspective, including: *anticipation*, *anger*, *fear*, *joy*, *disgust*, *love*, *optimism*, *sad*, *surprise*, *trust* and *pessimism*. RCECps contains 8 emotion labels: *anger*, *expectation*, *anxiety*, *joy*, *love*, *hate*, *sorrow* and *surprise*. There are totally 10,983 Chinese documents manually annotated from Chinese Weibo. After filtering out noise during pre-processing, we keep a total vocabulary size of 6.5 k for Sem18, and 7.3 k for RCECps.

B. Experimental Settings

For English, we use the publicly available GloVe[1] 300-dimensional embeddings trained on 6 billion words from Wikipedia and web text. For Chinese, we train 300-dimensional word embeddings on Chinese Wikipedia on 3.1 billion words using word2vec[2]. In our experiments, the topic module takes BoW as input, we thus filter out the stopword tokens[3]. In the learning process, we pre-train the emotion module and co-train the entire part with a batch size of 16, both under early-stop strategy. To mitigate overfitting, we apply word embedding dropout and layer dropout with rates of 0.3 and 0.01, respectively. We also make use of the contextualized language model BERT [50], which are from the official *Base* version.[4] in English and Chinese, respectively. We use Adam [51] for the optimization with initial rate of 0.001. All experiments are conducted with a GTX 1080Ti GPU and 11 GB graphic memory. Our systems are built based on PyTorch framework[5].

We employ five widely used metrics for measuring the performance of multi-label classification, including **Hamming Loss** (**HL**), **Ranking Loss** (**RL**), **Micro F1** (**miF1**), **Macro F1** (**maF1**) and **Average Precision** (**AP**) [12], [18].

[1][Online]. Available: http://nlp.stanford.edu/projects/glove/

[2][Online]. Available: https://code.google.com/archive/p/word2vec/

[3]For Chinese, we perform word segmentation first, by using the gensim package: [Online]. Available: https://radimrehurek.com/gensim/parsing/preprocessing.html

[4][Online]. Available: https://github.com/google-research/bert

[5][Online]. Available: https://pytorch.org/

C. Baselines

We compare the proposed model with strong baseline systems, which can be divided into two classes: models for multi-label classification (MLC) and methods for multi-label emotion classification (MLEC).

MLC: We first employ Binary Relevance (BR) methods. Zhang *et al.* (2014) transform multi-label problem into several binary problems [12]. Following their settings, we compare three typical neural classifiers that are widely used for text classification, including BiLSTM [52], AttLSTM [53] and the CNN-based FastText [54]. Note that these models encode sentences and extract features for each emotion learning via RNN or CNN architecture. Besides, we retrofit Transformer[6] [55] and the capsule based CapNet model [38] into BR scheme as our baselines, because these two models have been proven very effective in mining the useful context clues for the classification. We also compare some other types of models designed for multi-label classification tasks. 1) ECC constructs classifier chains for multi-label classification [13]. 2) ML-KNN extends the idea of the KNN model by calculating the Bayesian conditional probability of each label and tagging the one with highest probability among K samples as the final label [12]. 3) MLLOC constructs the label correlations under local perspective [56]. 4) TMC improves the feature extraction of multi-label classification via convolutional layers [57]. **MLEC**: We consider the following methods for multi-label emotion classification, including EDL [17], JBNN [14], SGM [58], RERc [18], INN-RER [30] and DATN [26]. Compared with MLC, MLEC has stronger ability because they universally mine the emotion-relevant interactions between emotions. For example, EDL and RERc attempt to capture the emotion distribution, or relationships between emotions. RERc and INN-RER both incorporate the topic representation information when determining an emotion label. Besides, DATN improves the performance of the task by integrating external emotion-rich knowledge via transfer learning.

D. Development Experiments

We conduct experiments on the development set to explore optimal hyperparamters including the routing iterations r of LTAR and DR, and the latent topic number T. The results are shown in Figure 2. First, the topic attention based routing algorithm takes more iterations with r $=5$ on both two datasets, compared with the dynamic routing with r $= 3$. This is reasonable since attention learning needs additional computation cost. Second, the best topic numbers on Sem18 and RCECps are 50 and 100, respectively, because the size of RCECps is larger than that of Sem18. By using a larger T, the performance drops quickly because of overfitting.

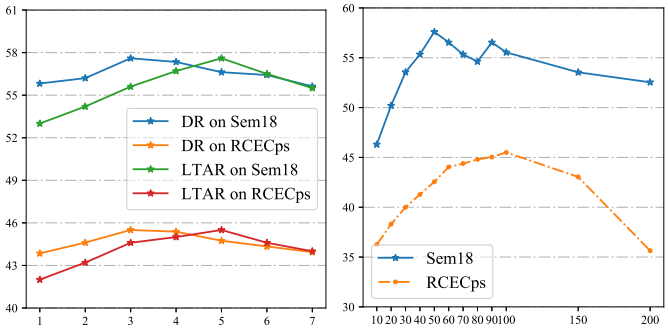
E. Main Results

The results of different models are shown in Table III. We have several observations. First, the MLEC models give better results

[6]The configuration keeps same with official Transformer version, and we only make use of the encoder (3-layer) for making classification.

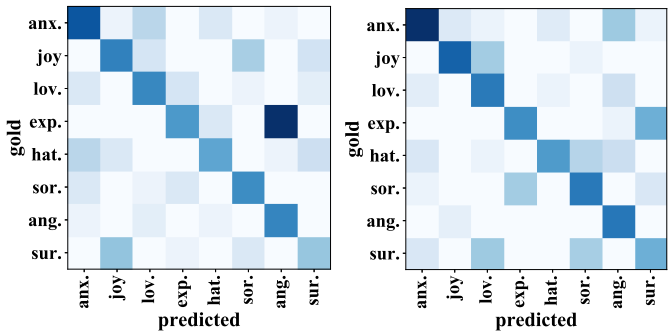
TABLE III
EXPERIMENTAL RESULTS OF DIFFERENT MODELS. W/O **LTAR** DENOTES REPLACING THE LATENT TOPIC ATTENTION BASED ROUTING WITH DYNAMIC ROUTING. W/O P^2 MEANS REMOVING THE FEATCAP LAYER. ↓ INDICATES SMALLER IS BETTER, AND ↑ INDICATES LARGER IS BETTER

	Sem18					RCECps				
	HL (↓)	**RL (↓)**	**miF1 (↑)**	**maF1 (↑)**	**AP (↑)**	**HL (↓)**	**RL (↓)**	**miF1 (↑)**	**maF1 (↑)**	**AP (↑)**
MLC:										
BiLSTM	0.245	0.344	0.498	0.437	0.400	0.212	0.370	0.290	0.277	0.582
AttLSTM	0.244	0.248	0.557	0.432	0.449	0.191	0.318	0.350	0.296	0.641
FastText	0.197	0.235	0.522	0.438	0.428	0.206	0.264	0.312	0.281	0.630
Transformer	0.185	0.191	0.590	0.505	0.502	0.191	0.224	0.451	0.376	0.653
CapNet	0.179	0.184	0.624	0.513	0.520	0.185	0.230	0.463	0.392	0.660
ECC	0.210	0.240	0.458	0.376	0.395	0.210	0.348	0.291	0.256	0.597
MLLOC	0.245	0.342	0.484	0.414	0.413	0.185	0.474	0.278	0.234	0.413
ML-KNN	0.196	0.270	0.410	0.387	0.391	0.245	0.290	0.310	0.285	0.591
TMC	0.191	0.219	0.561	0.465	0.482	0.228	0.252	0.391	0.342	0.630
MLEC:										
EDL	0.182	0.177	0.581	0.504	0.501	0.187	0.227	0.458	0.365	0.662
SGM	0.165	0.184	0.616	0.492	0.524	0.187	0.234	0.473	0.392	0.673
JBNN	0.190	0.192	0.632	0.528	0.526	0.165	0.192	0.418	0.380	0.693
RERc	0.176	0.170	0.651	0.539	0.530	0.201	0.210	0.511	0.416	0.683
INN-RER	0.165	0.174	0.641	0.546	0.542	0.172	0.188	0.520	0.424	0.679
DATN	-	-	-	0.551	-	-	-	-	0.441	0.732
TECap:										
Full	**0.141**	**0.152**	**0.682**	**0.576**	**0.579**	**0.152**	**0.178**	**0.531**	**0.455**	**0.754**
w/o **LTAR**	0.163	0.175	0.659	0.548	0.551	0.181	0.205	0.520	0.423	0.712
w/o P^2	0.170	0.180	0.657	0.550	0.549	0.192	0.213	0.523	0.438	0.720
w/o **LTAR** w/o P^2	0.185	0.197	0.620	0.515	0.523	0.211	0.245	0.489	0.402	0.679



(a) Routing iterations r (b) Topic numbers T

Fig. 2. Macro F1 over different settings.



(a) DATN (b) TECap

Fig. 3. Visualization of confusion matrix between TECap and DATN.

than the MLC methods, which demonstrates the importance to capture emotion-relevant information for the task. Second, among all MLC models, we find that Transformer and CapNet achieve better results than those vanilla encoders such as LSTM and CNN, and also outperform some typical methods designed for multi-label classification tasks, such as ML-KNN and TMC. The reason can be that both the self-attention mechanism in Transformer and the capsule architecture in CapNet can help capture the rich context information, facilitating the target emotion detection. Finally, we can see that our proposed model achieves the current best performance compared with all the baseline systems on almost all measurements, with 0.579 Average Precision, 0.141 Hamming Loss, 0.152 Ranking Loss on Sem2018, and 0.754 Average Precision, 0.152 Hamming Loss and 0.178 Ranking Loss on RCECps, respectively.

On the other hand, the models that integrate the additional knowledge, including RERc, INN-RER, DATN and TECap, can achieve better performance compared with the models without contextual or prior knowledge. Besides, the comparison between EDL and (RERc or INN-RER) show that the topic information is useful for the task. Here, EDL attempts to capture the emotion distribution between emotions for facilitating the final prediction, while RERc and INN-RER both incorporate the topic information when determining an emotion label. Finally, our model TECap with the capsule module outperforms the RNN/CNN based models by a large margin, demonstrating the ability of the capsule network on feature learning. The above analysis shows the effectiveness of our model for the task.

We also compare our model with strong baselines on each emotion. The results on the RCECps dataset are shown in Table IV. We can find that our model gives better results among all the emotion categories where the emotions **hate** and **surprise** are more challenging for all the models. One possible reason is that these two labels are relatively sparse in the dataset, while our model still gives better results than other models. In Figure 3, we further show the confusion matrices between our method and the

TABLE IV
RESULTS OF EACH EMOTION ON RCECPS. THE PERFORMANCES ARE MEASURED BY MACRO F1

Model	anxiety	joy	love	expectation	hate	sorrow	anger	surprise	Avg.
JBNN	0.507	0.456	0.446	0.380	0.268	0.442	0.304	0.251	0.380
EDL	0.453	0.402	0.399	0.259	0.255	0.366	0.307	0.239	0.335
RERc	0.557	0.505	0.453	0.407	0.320	0.444	0.332	0.290	0.352
DATN	0.598	0.481	0.464	0.413	0.376	0.450	0.470	0.276	0.441
TECap	**0.608**	**0.492**	**0.475**	**0.427**	**0.390**	**0.469**	**0.484**	**0.299**	**0.455**

TABLE V
RESULTS OF THE MODEL ABLATION. THE PERFORMANCE IS MEASURED BY MACRO F1

	Sem18	RCECp
TECap-Full	**0.576**	**0.455**
Topic Modeling		
w/o VAE	0.548(-0.028)	0.423(-0.032)
VAE†	0.551(-0.025)	0.434(-0.021)
LDA	0.556(-0.020)	0.439(-0.016)
Feature Encoding		
CNN	0.501(-0.075)	0.353(-0.102)
BiLSTM	0.485(-0.091)	0.361(-0.094)
Transformer	0.544(-0.032)	0.412(-0.043)
+BERT	0.597(+0.021)	0.470(+0.015)

best baseline DATN, in terms of each emotion, in order to better understand our model. By analyzing the incorrect predicted examples, we find that our model prefers to assign labels to the relevant emotions which may share more similarity under the closely related topics. In other words, our model is close enough to correctly predict these examples. For example, TECap predicts the emotion **anxiety** as the emotion **anger**, **expectation** as **surprise**, and **surprise** as **love**. The main reason is that VAE has strong ability in learning topic information. In contrast, such ability is not evident in other models such as DATN.

F. Ablation Study

From the results in Table III, we can find that, without topic information, the performance drops by 0.028 and 0.042 (AP) on the two datasets, respectively. This again shows the usefulness of the topic information for multi-label emotion detection. By removing FeatCap, the performance also drops by about the same amount as the topic information, demonstrating the strong ability of TECap on learning features. When both FeatCap and topic information are unavailable, the performance will drop quickly, but it is still better than most of the baselines. This shows the effectiveness of employing capsule network for multi-label classification.

We further explore the effects of the topic module on topic modeling, and the capsule module on feature extraction, respectively, which is shown in Table V. We first remove the topic model (VAE), and the capsule module gives 0.028 and 0.032 reduction, respectively. Then, we replace the well-trained topic model[7] with a sub-optimal VAE†, which is directly co-trained with the capsule module and stopped when the capsule network is convergent. We can see that such VAE is under-trained, and the performance consequently decreases, while it still better than that without VAE. If we use the topic information representation from LDA,[8] we can still obtain improvements which however are smaller than that of our topic module. For feature encoding, we use CNN, BiLSTM and Transformer for replacing the default, respectively. The input representation of these encoders are likewise concatenated with topic representation from the topic module, for fair comparisons. We can see that Transformer gives the best results compared with the other straightforward vanilla encoders (i.e., CNN and BiLSTM). Note that Transformer has an architecture with total stacked self-attention layers, which has been shown to be very effective on aggregating features [55], for each emotion label. Nevertheless, we see that the improvements in Transformer are not as obvious in our capsule module (0.032 and 0.043 reduction). This again validates the effectiveness of TECap in mining informative clues and refining effective features for each emotion type. Moreover, we show that with the help of the contextualized language model BERT, we can obtain boosted results for both two datasets, which coincides with the trends in recent studies about BERT.

V. DISCUSSION

A. Encapsulating Features via Capsule

As we mentioned earlier, we adopt the capsule network (CapNet) for feature encoding, because it allows encapsulated feature learning, which is more suitable for encoding the part-whole relationship for each separate emotion label in a multi-label scenario. By combining with dynamic routing, the learnt features for the corresponding emotion will be further iteratively refined. Here we verify such strength of CapNet. Following Yang *et al.* (2018) [38], we visualize the connection strength (the coupling coefficients) between *PrimCap Layer* and *EmoCap Layer*.[9] We also make comparison with the self-attention style Transformer model, on which we make attention visualization based on the last layer. We conduct experiments based on the sentence S3 in Table I.

As show in Figure 4, we can clearly observe the characteristic of CapNet on capturing the encapsulated features for each emotion separately. At the initial iteration, the prediction from CapNet is incomplete and biased. After 3 iterations, CapNet is able to refine the encapsulated clues and correctly predict the

[7]We use the pre-training technique to reach its upper bound for achieving the well-trained VAE.

[8]We first generate topic and keywords offline via LDA, and then let the model take as input such representations, the same way as in [28]

[9]To observe the direct projections between surface words (phrases) and emotion labels, we need to remove the middle *FeatCap layer*

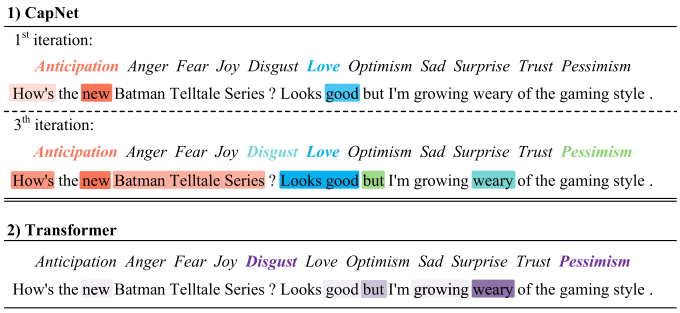


Fig. 4. Visualization of the CapNet mechanism, compared with Transformer model. Same color between emotion labels (top) and text words (bottom) indicates the identical projection. Deeper color indicates more contributions from the words (phrases).

TABLE VI
MACRO F1 OF CO-EXISTING EMOTIONS

Model	$\geq$ 1 co.	$\geq$ 2 co.	$\geq$ 3 co.
SGM	0.531	0.581	0.659
RERc	0.538	0.610	0.698
DATN	0.558	0.620	0.703
TECap	**0.582**	**0.651**	**0.740**

emotions assigned with proper intensities. Such observations are consistent with the previous findings on CapNet [38]–[41]. Compared with the capsule network, Transformer tends to retrieve clues for some homogenous types of emotions with overwhelming intensities, meanwhile suppressing those minority of signals which however can indicate other important emotion labels. For example, Transformer emphasize the '*weary*' and '*but*' clues with more weights than other words, consequently discovering merely the *Disgust* and *Pessimism* emotions while missing the *Anticipation* and *Love* emotions. This makes it more suitable for single-label classification, compared with CapNet.

B. Multi-Label Emotion Learning

We further analyze the ability on multi-label learning by changing the number of co-existing emotions. Table VI shows the results. Compared with SGM, The models RERc and DATN that leverage the external information achieve better performance. Note that our model gives the current best performance. We can also find that the more emotions co-exist, the more improvements TECap achieves. This indicates the ability of the proposed model in multi-label emotion detection.

C. Collaboration of Topic and Capsule Module

To explore how the topic module and the proposed topic attention based routing algorithm together help to make prediction, we show an example (S5 in Table I) from Sem18.

First, we visualize the coupling coefficients c_{ij} between FeatCaps and EmoCaps of the model trained by replacing **LTAR** with **DR**. From Figure 5(a), we can see that the capsules at the EmoCap layer correctly assigns higher weights to the emotions **anticipation**, **optimism** and **trust**. This is intuitively reasonable, since the capsule module can capture effective cues for supporting its prediction, such as *coming soon* and *I'll* thanks to

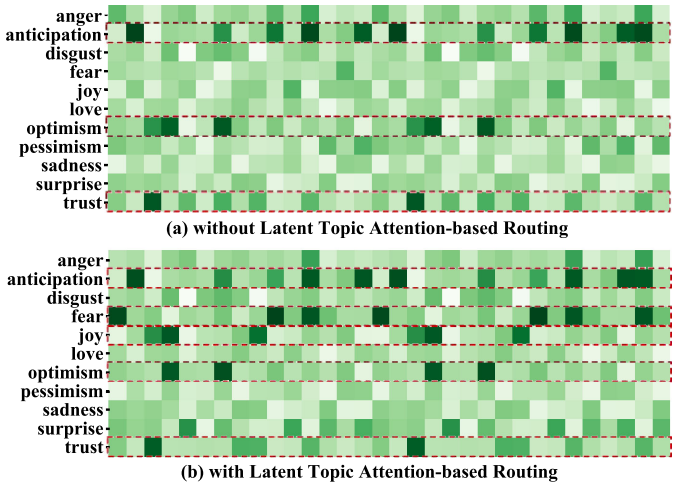


Fig. 5. Visualization of coupling coefficients c_{ij} between FeatCaps and EmoCaps without (a) and with (b) latent topic attention routing.

the routing mechanism. On the other hand, it is more effective to infer the **fear** and **joy** emotions when learning the latent topic information by involving the keyword ***Halloween***, in which many rich emotion-related keywords are actually retained.

Second, as show in Figure 5(b), TECap with the **LTAR** algorithm correctly assigns proper weights to the implicit emotions **fear** and **joy**. We further visualize the topic-keyword representation $\boldsymbol{M}$ (Eq.2) based on the same example, as illustrated in Figure 6, to analyze how the topic module help make inferences. First, four elements are highlighted in the latent variables $\boldsymbol{Z}$. These are the learnt topics which entail the corresponding keywords, respectively. In addition, the corresponding values to the token words of the sentence are highlighted correctly with the highly weighted topics. We can see that, for example, the tokens ***Halloween Party*** correspond to the 12-th topic, in which there are some keywords including *Halloween* and *witch*, showing the topic about *Halloween*, which can intuitively help the prediction of the **fear** emotion. Similarly, the 40-th topic with keywords including *great, happy* indicates a latent topic *Happy*, which provides the obvious evidences for supporting the **joy** emotion. The above analysis demonstrates the effectiveness of our model.

D. Topic Discovering

We print out the top5 keywords of the top2 topics in each emotion discovered by TECap on Sem18, as shown in Table VII. We can find that the keywords under the corresponding topics discovered by the topic module offer rich emotion-relevant contextual information for each corresponding emotion. This proves the effectiveness of the topic module. With such extended contextual information, our model is able to obtain better performance and more interpretable results.

E. Error Analysis

In this section, we present some cases, aiming to further analyze the limitations of our model. We mainly summarize two types of the incorrect predictions. The first involves deep semantic understanding of the texts, e.g., polysemy phenomenon.

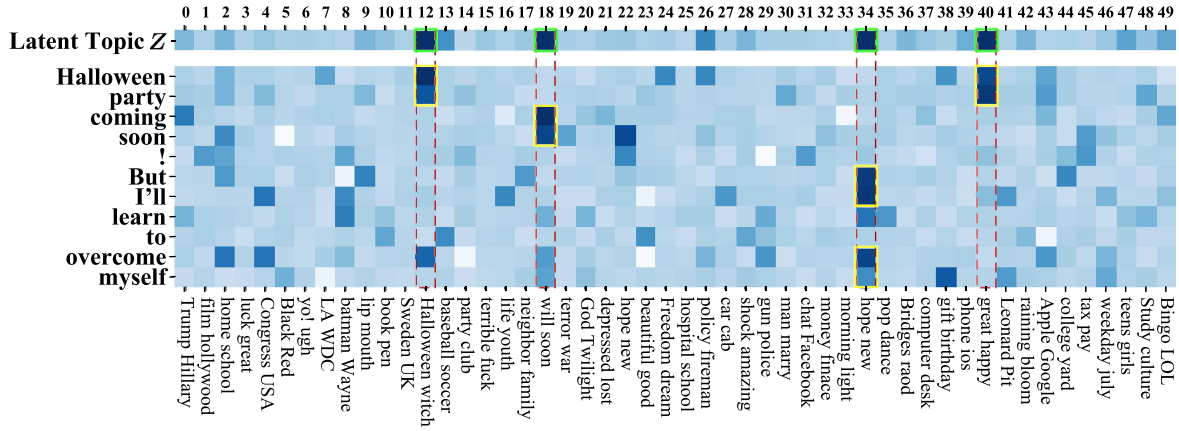


Fig. 6. Visualization of topic-keyword representation. The words at the bottom of the color map at the corresponding top2 keywords of each topic. The example sentence is shown on the left.

TABLE VII
TOP5 KEYWORDS OF TOP2 TOPICS IN EACH EMOTION

Emotion	Top5 Keywrods of Top2 Topics
anger	shocking bitter horrible awful angry
	Trump offense alarm Hillary gun
anticipation	Movie baby video films Series
	will soon wait new can't
disgust	terrible horrible fuming bully awful
	terrorism Trump Pakistan violence war
fear	Halloween witch terrify pumpkin ghost
	terror danger Iran attack dead
joy	great happy good amazing hilarious
	dance live.ly music party club
love	affection life youth feeling mood
	love happy amazing beautiful good
optimism	victory God twilight TED sunshine
	hope new smiling :) laughter
pessimism	lost depressing can't bad unhappy
	concern life end mood strange
sadness	depression lost unhappy can't nightmare
	die crime murder misfortune police
surprise	shock amazing hilarious serious frighten
	Snapchat Facebook chat twitters Youtube
trust	best new great faith smile
	Freedom dream hope home warmth

For example, the sentence "*That drill looks sick, I'm so loving it!*" reflects the emotion **love**, while TECap wrongly assigns an additional **disgust** emotion label with the keyword '*sick*'. Actually, '*sick*' in the sentence is a strong word for describing the emotion **love**. The reason is that the capsule network works by extracting n-gram features, the same way as CNN does. However, this nature of ignoring word order makes the capsule model easier to take certain words or phrase out of context, despite its ability and efficiency in feature learning.

The second type of errors comes from the topic modeling. In previous section, we show that the topic module is well capable of inducing topic information to support the learning of emotion relevance. Nevertheless, since VAE learns latent topic information unsupervisedly, the keywords under some different topics can overlap where may lead the model to make incorrect predictions. Taking the example in Figure 6, the top2 keywords in topic #22 and topic #34 are coincidently the same. Assuming that the topic #22 is the emotion **surprise** and the topic #22 is the emotion **anticipation**. If the topic information from #22 is used, the classifier will most likely predict a **surprise** label, even though the sentence S5 does not contain such emotion.

VI. CONCLUSION

We proposed a topic-enhanced capsule network for multi-label emotion classification, which could learn the latent topic information without external resources, effectively leveraging it into a capsule-based classifier for multiple emotions prediction. Results on two benchmark datasets showed that our method outperformed strong baselines by a large margin, demonstrating the effectiveness of capturing topic information and learning rich features for the task. Further in-depth analysis revealed the strengths and limitations of the topic module and capsule module in our proposed model.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their detailed comments, which have helped us to improve the quality of this work.

REFERENCES

[1] J. Xu, R. Xu, Q. Lu, and X. Wang, "Coarse-to-fine sentence-level emotion classification based on the intra-sentence features and sentential context," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 2455–2458.
[2] Z. Wang, S. Y. M. Lee, S. Li, and G. Zhou, "Emotion analysis in code-switching text with joint factor graph model," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 25, no. 3, pp. 469–480, Mar. 2017.
[3] Y. Ren, Y. Zhang, M. Zhang, and D. Ji, "Improving Twitter sentiment classification using topic-enriched multi-prototype word embeddings," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 3038–3044.
[4] X. Quan, Q. Wang, Y. Zhang, L. Si, and L. Wenyin, "Latent discriminative models for social emotion detection with emotional dependency," *ACM Trans. Inf. Syst.*, vol. 34, no. 1, pp. 1–2, 2015.
[5] G. Yang, H. He, and Q. Chen, "Emotion-semantic-enhanced neural network," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 27, no. 3, pp. 531–543, Mar. 2019.

[6] H. Zhou, M. Huang, T. Zhang, X. Zhu, and B. Liu, "Emotional chatting machine: Emotional conversation generation with internal and external memory," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 730–738.

[7] K. J. Oh, D. Lee, B. Ko, and H. J. Choi, "A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation," in *Proc. 18th IEEE Int. Conf. Mobile Data Manage.*, 2017, pp. 371–375.

[8] T. H. Nguyen, K. Shirai, and J. Velcin, "Sentiment analysis on social media for stock movement prediction," *Expert Syst. Appl.*, vol. 42, no. 24, pp. 9603–9611, 2015.

[9] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and lstm," *Neural Comput. Appl.*, vol. 31, no. 3, pp. 1–17, 2019.

[10] A. Bermingham and A. Smeaton, "On using Twitter to monitor political sentiment and predict election results," in *Proc. Workshop Sentiment Anal. AI Meets Psychol.*, 2011, pp. 2–10.

[11] J. I. Park and T. Kim, "Improving policies and regulations for environmental-friendly ocean renewable energy development in korea," *Decis. Support Syst.*, vol. 23, no. 4, pp. 237–250, 2014.

[12] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[13] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2009, pp. 254–269.

[14] H. He and R. Xia, "Joint binary neural network for multi-label learning with applications to emotion classification," in *Proc. Natural Lang. Process. Chin. Comput.*, 2018, pp. 250–259.

[15] W. Ying, R. Xiang, and Q. Lu, "Improving multi-label emotion classification by integrating both general and domain-specific knowledge," in *Proc. 5th Workshop Noisy User-Generated Text*, 2019, pp. 316–321.

[16] S. AlZu'bi, O. Badarneh, B. Hawashin, M. Al-Ayyoub, N. Alhindawi, and Y. Jararweh, "Multi-label emotion classification for Arabic tweets," in *Proc. 6th Int. Conf. Soc. Netw. Anal., Manage. Secur.*, 2019, pp. 499–504.

[17] D. Zhou, X. Zhang, Y. Zhou, Q. Zhao, and X. Geng, "Emotion distribution learning from texts," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 638–647.

[18] D. Zhou, Y. Yang, and Y. He, "Relevant emotion ranking from text constrained with emotion relationships," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, 2018, pp. 561–571.

[19] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse Processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[20] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[22] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," 2015, *arXiv:1511.06349*.

[23] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *Proc. 33rd International Conf. Int. Conf. Mach. Learn.*, 2016, pp. 1727–1736.

[24] Z. Chen and T. Qian, "Transfer capsule network for aspect level sentiment classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguist.*, 2019, pp. 547–556.

[25] X. Chen, C. Lyu, and I. Titov, "Capturing argument interaction in semantic role labeling with capsule networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 5415–5425.

[26] J. Yu, L. Marujo, J. Jiang, P. Karuturi, and W. Brendel, "Improving multi-label emotion classification via sentiment classification with dual attention transfer network," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1097–1102.

[27] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Proc. NeurIPS*, 2015, pp. 919–927.

[28] Y. Ren, Y. Zhang, M. Zhang, and D. Ji, "Context-sensitive twitter sentiment classification using neural network," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 215–221.

[29] B. Gretarsson, J. ODonovan, S. Bostandjiev, T. Hllerer, and P. Smyth, "Topicnets: Visual analysis of large text corpora with topic modeling," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 2, pp. 1–26, 2012.

[30] Y. Yang, Z. Deyu, and Y. He, "An interpretable neural network with topical information for relevant emotion ranking," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3423–3432.

[31] H. Bahuleyan, L. Mou, O. Vechtomova, and P. Poupart, "Variational attention for sequence-to-sequence models," 2017, *arXiv:1712.08207*.

[32] W. Aziz and P. Schulz, "Variational inference and deep generative models," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguist.*, 2018, pp. 8–9.

[33] H. Fei, Y. Ren, and D. Ji, "Implicit objective network for emotion detection," in *Proc. Natural Lang. Process. Chin. Comput.*, 2019, pp. 647–659.

[34] Y. Li, T. Baldwin, and T. Cohn, "Semi-supervised stochastic multi-domain learning using variational inference," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguist.*, 2019, pp. 1923–1934.

[35] X. Zhang, Y. Yang, S. Yuan, D. Shen, and L. Carin, "Syntax-infused variational autoencoder for text generation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguist.*, 2019, pp. 2069–2078.

[36] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 44–51.

[37] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. NeurIPS*, 2017, pp. 3856–3866.

[38] M. Yang, W. Zhao, J. Ye, Z. Lei, Z. Zhao, and S. Zhang, "Investigating capsule networks with dynamic routing for text classification," in *Proc.Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3110–3119.

[39] Z. Yang, J. Zhang, F. Meng, S. Gu, Y. Feng, and J. Zhou, "Enhancing context modeling with a query-guided capsule network for document-level translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 1527–1537.

[40] C. Du *et al.*, "Capsule network with interactive attention for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 5489–5498.

[41] M. Wang, "Towards linear time neural machine translation with capsule networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 803–812.

[42] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Investigating dynamic routing in tree-structured LSTM for sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 3432–3437.

[43] H. Liu *et al.*, "Reconstructing capsule networks for zero-shot intent classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2019, pp. 4799–4809.

[44] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proc. World Wide Web Conf.*, 2018, pp. 1165–1174.

[45] N. Zhang, S. Deng, Z. Sun, X. Chen, W. Zhang, and H. Chen, "Attention-based capsule networks with dynamic routing for relation extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 986–992.

[46] A. G. A. P. Goyal, A. Sordoni, M.-A. Côté, N. R. Ke, and Y. Bengio, "Z-forcing: Training stochastic recurrent networks," in *Proc. NeurIPS*, 2017, pp. 6713–6723.

[47] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.

[48] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "Semeval-2018 task 1: Affect in tweets," in *Proc. SemEval*, 2018, pp. 1–17.

[49] C. Quan and F. Ren, "Sentence emotion analysis and recognition based on emotion words using ren-cecps," *Int. J. Adv. Intell.*, vol. 2, no. 1, pp. 105–117, 2010.

[50] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist.: Human Lang. Technol.*, 2019, pp. 4171–4186.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[52] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[53] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguist.*, 2016, pp. 207–212.

[54] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," 2016, *arXiv:1607.01759*.

[55] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.

[56] S.-J. Huang and Z.-H. Zhou, "Multi-label learning by exploiting label correlations locally," in *Proc. 36th AAAI Conf. Artif. Intell.*, 2012, pp. 949–955.

[57] Y. Wang, S. Feng, D. Wang, G. Yu, and Y. Zhang, "Multi-label Chinese microblog emotion classification via convolutional neural network," in *Proc. APWeb*, 2016, pp. 567–580.

[58] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "Sgm: sequence generation model for multi-label classification," in *Proc. 27th Int. Conf. Comput. Linguist.*, 2018, pp. 3915–3926.