Learning Discriminative Neural Sentiment Units for Semi-supervised Target-Level Sentiment Classification

Jingjing Zhao[1], Yao Yang[1](✉), Guansong Pang[2], Lei Lv[1], Hong Shang[1], Zhongqian Sun[1], and Wei Yang[1]

[1] Tencent AI Lab, Shenzhen, China
{laceyzhao,yaoyang,lvleilv,hongshang,sallensun,willyang}@tencent.com
[2] Australian Institute for Machine Learning,
University of Adelaide, Adelaide, Australia
pangguansong@gmail.com

Abstract. Target-level sentiment classification aims at assigning sentiment polarities to opinion targets in a sentence, for which it is significantly more challenging to obtain large-scale labeled data than sentence/document-level sentiment classification due to the intricate contexts and relations of the target words. To address this challenge, we propose a novel semi-supervised approach to learn *sentiment-aware representations* from easily accessible unlabeled data specifically for the fine-grained sentiment learning. This is very different from current popular semi-supervised solutions that use the unlabeled data via pretraining to generate *generic representations* for various types of downstream tasks. Particularly, we show for the first time that we can learn and detect some highly sentiment-discriminative neural units from the unsupervised pretrained model, termed *neural sentiment units*. Due to the discriminability, these sentiment units can be leveraged by downstream LSTM-based classifiers to generate sentiment-aware and context-dependent word representations to substantially improve their sentiment classification performance. Extensive empirical results on two benchmark datasets show that our approach (i) substantially outperforms state-of-the-art sentiment classifiers and (ii) achieves significantly better data efficiency.

Keywords: Discriminative neural sentiment units · Target-level sentiment analysis · Deep neural network

1 Introduction

Target-level sentiment classification (TSC) is the task of classifying sentiment polarities on opinion targets in sentences. It can provide more detailed insights into sentence polarities, but it involves significantly more intricate sentiment relations than sentence/document-level sentiment analysis. For example, the sentence "*The voice quality of this phone is not good, but the battery life is long*"

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12085, pp. 798–810, 2020.
https://doi.org/10.1007/978-3-030-47436-2_60

holds negative sentiment on the target "*voice quality*" but is positive on the target "*battery life*".

In recent years, deep neural network-based methods have been extensively explored for target-level sentiment classification to learn the representations of sentences and/or targets. Recurrent neural networks are one of the most popular approaches for this task because of their strong capability of learning sequential representations [2,9].

However, these methods fail to distinguish the importance of each word to the target. A range of attention mechanisms are introduced to address this issue, such as target-to-sentence attention [2], fine-grained word-level attention [3], and multiple attentions [4]. Convolutional neural network (CNN)-based models are also recently used for this task because of the capability to extract the informative n-grams features [5]. All the aforementioned methods focus on exploiting *labeled data* to build the classification model, whose performance is often largely limited. This is because they normally require large-scale high-quality labeled data to be well trained, but in practice we have only small target-level labeled data since it is very difficult and costly to collect due to the complex nature of the task, e.g., fine granularity, co-existence of multiple targets in a sentence, and context-sensitive sentiment. Two main methods to address this issue include: (i) generating and incorporating extra sentiment-informative representations by using auxiliary knowledge resources, e.g., sentiment lexicons [17,28]; and (ii) pretraining the embeddings of words or the parameters of networks using large-scale unlabeled data [3,16]. However, both methods can't capture context-dependent sentiment. For example, the opinion "***long***" can have completely opposite sentiment in different contexts, e.g., it is positive in " *battery life is* ***long***" but negative in "*the start-up time is too* ***long***". Additionally, the sentiment lexicons require very expensive human involvement to handle data with evolving and highly diversified linguistics, so the pretraining method is more plausible.

The pretraining aims at generating *generic* representations for different learning tasks, which can often extract some transferable features for a particular task. However, due to the generic learning objective, it can also extract a large number of features that are irrelevant or even noisy w.r.t. a given task such as sentiment classification, leading ineffective use of the unlabeled data. In this study, we introduce a novel approach to associate the feature learning on unlabeled data with the downstream sentiment classification to extract highly relevant features w.r.t. sentiment classification. Specifically, besides pretraining on unlabeled data, we take a step further to learn and extract highly sentiment-discriminative neural units from a pretrained model, e.g., long short-term memory (LSTM)-based Variational Autoencoder (VAE) [11]. The selective sentiment-aware units, termed Neural Sentiment Units (NeSUs), can generate highly relevant sentiment-aware representations, which are then leveraged by LSTM networks to perform sentiment classification on small labeled data. This enables LSTM networks to achieve significantly improved data efficiency and to learn context-dependent sentiment representations, resulting in substantially improved LSTM networks. In summary, this paper makes the following two main contributions:

- We discover for the first time that feature learning on unlabeled data can be associated with downstream sentiment classification to learn some highly sentiment-discriminative neural units (NeSUs). These NeSUs can be leveraged by LSTM-based classifiers to generate sentiment-aware and context-dependent representations, carrying substantially more task-dependent information than the generic representations obtained by pretraining.
- We further propose a novel LSTM-based target-level sentiment classifier called NeaNet that effectively incorporates the most discriminative NeSU to exemplify the applications of the NeSUs. Extensive empirical results on two benchmark datasets show that NeaNet (i) substantially outperforms 13 (semi-) supervised state-of-the-art sentiment classifiers and (ii) achieves significantly better data efficiency.

2 Related Work

Many methods have been introduced for target-level sentiment analysis, including rule-based approaches [1,6], statistical approaches [7,8] and deep approaches [9,21]. Due to page limits, below we discuss two closely relevant research lines.

Deep Methods. Recursive neural network is one popular network architecture explored at the early stage [29], which heavily relies on the effectiveness of syntactic parsing tree. Recurrent neural networks have also shown expressive performance in this task. TD-LSTM [9] incorporated target information into LSTM and modeled preceding and following contexts of the target to boost the performance. Target-sensitive memory networks (TMNs) [21] were proposed to capture the sentiment interaction between targets and contexts to address the context-dependent sentiment problem. However, these models fail to identify the contribution of each word to the targets. The attention mechanism [2,4,10,22] is then applied to address this issue. For example, A target-to-sentence attention mechanism, ATAE-LSTM [2], was introduced to explore the connection between the target and its context; IARM [22] leveraged recurrent memory networks with multiple attentions to generate target-aware sentence representations. As CNN can capture the informative n-grams features, convolutional memory networks were explored in [18] to incorporate an attention mechanism to sequentially compute the weights of multiple memory units corresponding to multi-words. Instead of attention networks, [5] proposed a component to generate target-specific representations for words, and employed a CNN layer as the feature extractor relying on a mechanism of preserving the original contextual information. Some other works [20] exploited human reading cognitive process for this task. These neural network-based methods stand for the current state-of-the-art techniques, but their performance are generally limited by the amount of high-quality labeled data.

Semi-supervised Methods. Many semi-supervised methods have been explored on sentence-level sentiment classification, such as pretraining with Restricted Boltzmann Machine or autoencoder [23,26], auxiliary task learning [24] and adversarial training [25,27]. However, there are only few studies [16,19] on semi-supervised target-level sentiment classification. [19] explored both pretraining and multi-task learning for transferring knowledge from document-level data, which is much less expensive to obtain. [16] used a Transformer-based VAE for pretraining, which modeled the latent distributions via variational inference. However, it failed to distinguish the relevant and irrelevant features with respect to the sentiment.

3 Neural Sentiment Units-Enabled Target-Level Sentiment Classification

3.1 The Proposed Framework

We introduce a novel semi-supervised framework to learn sentiment-discriminative neural units (NeSUs) on large-scale unlabeled data to enhance downstream classifiers on small labeled data. Unlike the widely-used pretraining approaches that learn generic representations, our proposed approach is specifically designed for fine-grained sentiment classification, by incorporating sentiment-aware neural units hidden in the pretrained model into downstream LSTM-based classifiers. This enables us to have a substantially more effective use of the unlabeled data, greatly lifting the sentiment classification on limited labeled data.

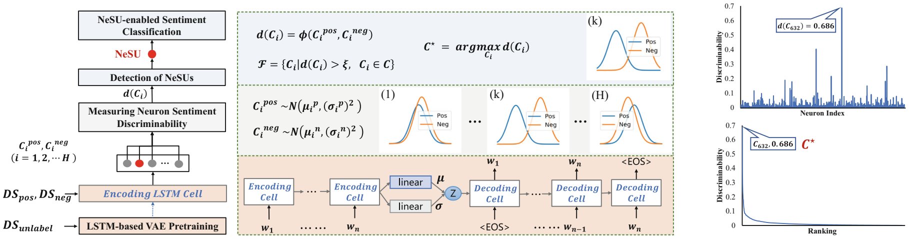
The procedure of our framework is presented in Fig. 1, which consists of four modules, including LSTM-based VAE pretraining, measuring neuron sentiment discriminability, detection of NeSUs, and NeSU-enabled sentiment classification. The details of each module are introduced below.

3.2 LSTM-Based VAE Pretraining

VAE is composed of an encoder and a decoder. The encoder maps an input $\mathbf{x}$ into a latent space and outputs the representation $\mathbf{z}$. The decoder decodes $\mathbf{z}$ to generate the input $\mathbf{x}$. LSTM-based VAE is used to pretrain for two main reasons: (i) VAE retains sentiment-related features which are important to generate sentences. (ii) LSTMs use an internal memory to remember semantic information, which can help learn intricate context-dependent opinions in sentiment analysis. VAE is trained on unlabeled data $DS_{unlabel}$ by minimizing reconstruction loss and KL divergence loss. And we obtain H neuron units for the encoding/decoding stage. We then exploit small labeled data to examine the discriminability of each neuron unit as follows.

3.3 Measuring Neuron Sentiment Discriminability

Definition 1 (Neuron Discriminability). *Let* $DS_{pos} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M\}$ *be the sentence set with positive sentiment and* $DS_{neg} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_K\}$ *be the*



(a) The proposed NeSU-enabled TSC framework (b) Neuron Discriminability

Fig. 1. The proposed NeSU-enabled target-level sentiment classification framework. Firstly, an LSTM-based VAE is trained on unlabeled data $DS_{unlabel}$. We then evaluate the discriminability of each encoding LSTM neuron unit using labeled data. A distribution separation measure $d(\cdot)$ is further applied to find a set of NeSUs ($\mathcal{F}$) that have the best discriminability. Since NeSUs are often redundant to each other, only the most discriminative NeSU ($C^\star$) is leveraged by the downstream classifiers.

sentences with negative sentiment, then we define the discriminability measure function $d(\cdot)$ w.r.t. a neuron unit C_i as follows:

$$d(C_i) = \phi(\eta_i(DS_{pos}), \eta_i(DS_{neg})), \tag{1}$$

where $\eta_i : DS \mapsto \mathbb{R}^{M+K}$ returns a vector that contains the last hidden states of the neuron unit C_i for all the sentences in the set $DS = \{DS_{pos}, DS_{neg}\}$, i.e., for M positive sentences and K negative sentences; the unit C_i has a scalar output; $\phi(\cdot,\cdot)$ is a measure that evaluates the separability of hidden states' distributions resulted by the samples of the two classes.

The main intuition of Definition 1 is that if a neuron unit has good discriminability, its hidden state distributions of different classes' samples should be well separable. Motivated by the fact that Gaussian distribution is the most general distribution for fitting values drawn from Gaussian/non-Gaussian variables according to the central limit theorem, we specify ϕ using *Bhattacharyya distance* to measure the separability of two distributions, which assumes the resulting hidden states in the neuron unit C_i for each class's samples follow a Gaussian distribution. Accordingly, the discriminability of C_i is calculated as follows:

$$\phi\left(C_i^{pos}, C_i^{neg}\right) = \frac{1}{4}\ln\left(\frac{1}{4}\left(\frac{({\sigma_i}^p)^2}{({\sigma_i}^n)^2} + \frac{({\sigma_i}^n)^2}{({\sigma_i}^p)^2} + 2\right)\right) + \frac{1}{4}\left(\frac{\left({\mu_i}^p - {\mu_i}^n\right)^2}{\left({\sigma_i}^p\right)^2 + \left({\sigma_i}^n\right)^2}\right), \tag{2}$$

where $C_i^{pos} \sim \mathcal{N}(\mu_i^p, (\sigma_i^p)^2)$ contains the hidden state values of C_i w.r.t. all the sentences with positive polarity; Similarly, $C_i^{neg} \sim \mathcal{N}(\mu_i^n, (\sigma_i^n)^2)$ contains the hidden state values for the negative polarity. A larger ϕ indicates greater separability between two hidden state distributions, thus, better discriminability.

3.4 Detection of Neural Sentiment Units (NeSUs)

Definition 2 (Neural Sentiment Units). *Let $\mathcal{C} = \{C_1,\ C_2 \ldots, C_H\}$ be the encoding LSTM neural unit set. Then neural sentiment units are defined as the neuron units with significantly large discriminability values:*

$$\mathcal{F} = \{C_i |\ d\,(C_i)\ >\ \xi,\quad C_i \in \mathcal{C}\}\,, \tag{3}$$

where ξ is a threshold hyperparameter and $\mathcal{F}$ is a set of discriminative NeSUs. Since each NeSU is an LSTM neural unit, it works as a none-linear mapping function $\eta : \mathbb{R}^D \mapsto \mathbb{R}$ which is the same η as Eq. 1 and can be formally defined as follows:

$$s_t = \eta(\mathbf{v}_t), \tag{4}$$

where $\mathbf{v}_t$ is an embedding vector of the t-th word and s_t is a scalar sentiment indication value with larger s_t indicating more positive sentiment.

In Fig. 1(b), we illustrate the discriminability values of all encoding LSTM neural units on a dataset `Laptop`. It is clear that only a small number of neural units are sentiment-aware. Most units do not capture much sentiment information. Therefore, simply using all units may disregard discriminative information. Instead, as defined in Eq. (3), we only retain selective sentiment-aware neural units based on their discriminability to fully exploit the unlabeled data.

The parameter ξ can be tuned via cross validation using the labeled data. We find that retaining the single most discriminative neural sentiment unit (NeSU) always results in the best downstream classification performance; adding more NeSUs does not perform better. This demonstrates that NeSUs in $\mathcal{F}$ capture similar transferable features, so they are often redundant to each other. We therefore only extract NeSU below for the downstream classification:

$$C^\star = \underset{C_i \in \mathcal{F}}{\arg\max}\, d\,(C_i), \tag{5}$$

where the unit $C^\star$, denoted by $\eta^\star$, is the only neural sentiment unit incorporated into downstream classifiers.

3.5 NeSU-Enabled LSTMs for Sentiment Classification

We further introduce a novel NeSU-enabled attention Network, namely NeaNet, by using two parallel LSTMs to fully exploit the NeSU and generate sentiment-aware representations for target-level sentiment classification.

Task Statement. The target-level sentiment analysis is to predict a sentiment category for a (sentence, target) pair. Given a sentence-target pair $\mathbf{x} = (\mathbf{w}, \mathbf{w}^T)$, where $\mathbf{w} = \{\mathbf{w}_1, \mathbf{w}_2\ , \ldots, \mathbf{w}_n\}$, $\mathbf{w}^T = \left\{\mathbf{w}_1^T, \mathbf{w}_2^T\ , \ldots, \mathbf{w}_m^T\right\}$, and $\mathbf{w}^T$ is a subsequence of $\mathbf{w}$. The goal of this task is to predict a sentiment polarity $y \in \{P, N, O\}$ of the sentence $\mathbf{w}$ w.r.t. the target $\mathbf{w}^T$, where P, N, and O denote "positive", "negative" and "neutral" sentiments respectively.

The architecture of NeaNet is shown in Fig. 2. The bottom is an embedding layer, which maps the words in an input sequence $\mathbf{w}$ to a word vectors

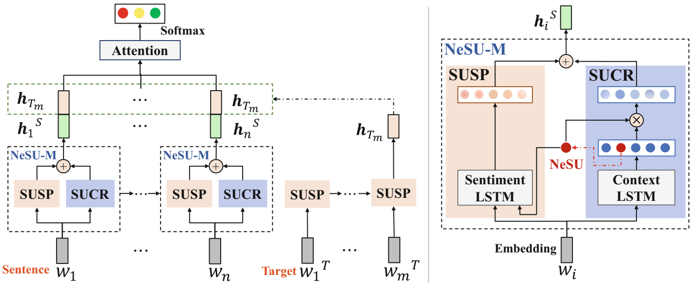


Fig. 2. The framework of NeaNet. SUSP and SUCR are the two NeSU-driven modules (NeSU-M). $\mathbf{h}_i^S$ is the integrated word representation of SUSP and SUCR, which carries context-dependent sentiments w.r.t. the target $\mathbf{h}_{T_m}$.

$\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ according to an embedding lookup table $\mathbb{L} \in \mathbb{R}^{D \times V}$ generated by the pretrained VAE, where D is the dimension of word vectors and V is the vocabulary size. The middle part consists of two core components which exploit NeSU to generate sentiment-aware representations, namely NeSU as Sentiment Prior (SUSP) and NeSU as Context Reinforcer (SUCR). The top parts are an attention layer and a softmax layer to combine the dual NeSU-driven modules to extract informative features for classification.

SUSP: Using NeSU as Sentiment Prior. Since NeSU can discriminate the sentiment of the input words, we integrate it into the memory computation of LSTM to generate sentiment-aware word representations. Moreover, the sentiment information can be carried forward along with word sequences due to the LSTM structure. Besides the three gates (input, forget and output gates) in the vanilla LSTM, we define an additional read gate $r_t \in [0, 1]$ to control the sentiment information captured by the NeSU $\eta^\star$. This yields a NeSU-enabled Sentiment LSTM. The NeSU works like a sentiment prior, so we call the whole module NeSU-based Sentiment Prior (SUSP), which is defined as follows:

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{v}_t + \mathbf{U}_i \mathbf{h}_{t-1}), \qquad \mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{v}_t + \mathbf{U}_f \mathbf{h}_{t-1}), \tag{6}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{v}_t + \mathbf{U}_o \mathbf{h}_{t-1}), \qquad \widehat{\mathbf{c}}_t = tanh(\mathbf{W}_c \mathbf{v}_t + \mathbf{U}_c \mathbf{h}_{t-1}), \tag{7}$$

$$\underline{r_t = \sigma(\mathbf{W}_d(\mathbf{W}_r \mathbf{v}_t + \ U_{\mathbf{r}} \mathbf{h}_{t-1}))}, \qquad \underline{d_t = r_t * s_t}, \tag{8}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \widehat{\mathbf{c}}_t + \underline{tanh(d_t \mathbf{z}_{su})}, \qquad \mathbf{h}_t = \mathbf{o}_t \odot \mathbf{c}_t, \tag{9}$$

where σ refers to sigmoid activation function and $tanh$ refers to hyperbolic tangent function; $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t \in \mathbf{R}^H$ respectively denote the input, forget and output gates; $\mathbf{v}_t$ is the t-th word embedding and $\mathbf{h}_{t-1}$ is the hidden state at time step $t-1$; $\mathbf{W}_i$, $\mathbf{W}_f$, $\mathbf{W}_o$, $\mathbf{W}_r$, $\mathbf{W}_c \in \mathbb{R}^{H \times D}$, $\mathbf{U}_i$, $\mathbf{U}_f$, $\mathbf{U}_o$, $\mathbf{U}_r$, $\mathbf{U}_c \in \mathbb{R}^{H \times H}$, $\mathbf{W}_d \in \mathbb{R}^{1 \times H}$ and $\mathbf{z}_{su} \in \mathbb{R}^H$ are the network weights, where H is the number of hidden cells; $s_t = \eta^\star(\mathbf{v}_t)$ denotes the sentiment value output by the retained NeSU mapping function $\eta^\star$ as in Eq. (4); $\odot$ denotes element-wise multiplication.

Table 1. Basic statistics of datasets and settings of hyperparamters.

Labeled data					Unlabeled data				Hyper-parameters	Laptop	Rest.
Dataset		#Positive	#Negative	#Neutral	Dataset		#Sample	#Total	LSTM dropout	0.5	–
Laptop	Train	980	858	454	Review	Laptop	38,742	379,813	Embedding dropout	0.5	0.5
	test	340	128	171		Rest.	119,822		Batch size	64	25
Rest.	Train	2159	800	632		Elec.	221,249		Attention size	50	50
	Test	730	195	196					$D/H/C$	512/1024/40	

Essentially, SUSP uses the NeSU $\eta^\star$, via the underlined parts in Eq. (8–9) to capture context-dependent sentiment information and propagate this information to generate the context-dependent representation $\mathbf{h}_t$.

The position information between the target and its context is also used to weight opinion words. The position weight l_i of $\mathbf{w}_i$ is calculated as follows:

$$l_i = \begin{cases} 1 - \frac{k-i}{C}, & i < k \\ 1, & k \le i \le k+m \\ 1 - \frac{i-(k+m)}{C}, & i > k+m \end{cases} \tag{10}$$

where k is the index of the first target word, m is the length of the target, and C is a constant associated with datasets. Finally $\mathbf{h}_t$ is weighted with l_t as:

$$\widetilde{\mathbf{h}}_t = \mathbf{h}_t * \ l_t. \tag{11}$$

SUCR: Using NeSU as a Context Reinforcer. Due to the integrated computation of Sentiment LSTM, some original context information might be lost. To preserve the genuine context, we parallelly employ a Context LSTM initialized with the VAE encoder to learn the generic word representation, and further incorporate NeSU with the position l to sentimentally reinforce the context representations generated by the Context LSTM. We call this whole module NeSU-based Context Reinforcer (SUCR) and define it as follows:

$$\widetilde{\mathbf{h}}_{e_t} = \mathbf{h}_{e_t} * |\mathrm{s}_t| * l_t, \tag{12}$$

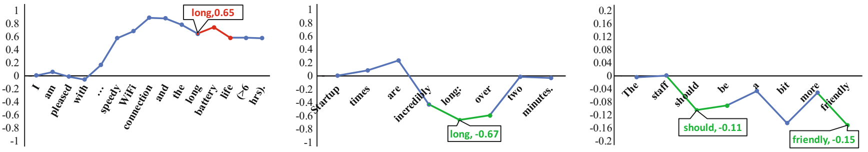
where $\mathbf{h}_{\mathbf{e}t}$ is the hidden state generated by the Context LSTM at the t-th time step and s_t is a sentiment value output by $\eta^\star$ as in Eq. (4).

Dual LSTMs for Classifying Target Sentiment. We further consolidate the word-level representations generated by SUSP and SUCR via summation to form the final sentiment-aware and context-sensitive word representations. Then we apply a standard attention layer to fuse the semantic information of the context and the target. Particularly, let $\mathbf{h}_{T_m}$ be the target representation generated by SUSP, $\widetilde{\mathbf{h}}_t$ and $\widetilde{\mathbf{h}}_{e_t}$ respectively denote the word representations generated by SUSP and SUCR. The input of attention layer is given as: $[\widetilde{\mathbf{h}}_t + \widetilde{\mathbf{h}}_{e_t}; \mathbf{h}_{T_m}]$.

4 Experiments

4.1 Experimental Settings

We evaluate our method on two benchmark datasets: `Laptop` and `Rest` from SemEval 2014 [30], containing reviews in laptop and restaurant domains. Following previous works [4,5], we remove the samples labeled "conflict". For VAE pretraining, a relatively large unlabeled dataset was collected, including Laptop, Rest. and Elec.. The unlabeled data Laptop and Rest. are respectively obtained from the Amazon Product Reviews[1] and Kaggle[2], while Elec. is from [14]. The statistics of all datasets and the detailed hyperparameters are listed in Table 1. For both labeled and unlabeled data, any punctuation is treated as space.



(a) 'long' is positive (b) 'long' is negative (c) a subjunctive sentence

Fig. 3. Visualization of the NeSU value for each word, as defined in Eq. (4). The red/green lines are to highlight where positive/negative sentiment concentrates. (Color figure online)

4.2 Visualizing and Understanding NeSU

To understand the discriminability of NeSU, this section demonstrates the sentiment NeSU perceives for each word in different sentences. It is clear that NeSU responds to the sentiment word "***long***" adaptively depending on the context, i.e., it is positive in Fig. 3(a) and negative in Fig. 3(b). In Figs. 3(a), benefiting from the LSTM, the target "***battery life***" can arouse the NeSU memory from "***long***", generating a higher value. Fig. 3(c) shows an example with subjunctive style, a challenging task for [5]. The NeSU can correctly assign a negative value for the positive sentiment word "***friendly***", and a downtrend/uptrend for "***bit***"/"***more***", demonstrating NeSU is also aware of implicit semantics.

4.3 Comparison to State-of-the-Art Methods

Overall Performance. The results are shown in Table 2. On both datasets, our model NeaNet consistently achieves the best performance in both accuracy (ACC) and macro-F1 compared to all 13 supervised and semi-supervised methods. E.g., compared to RAM, MGAN, TNet and ASVAET, which are the

[1] http://times.cs.uiuc.edu/~wang296/Data/.
[2] https://inclass.kaggle.com/c/restaurant-reviews.

Table 2. Results of all models on two benchmark datasets. The top two performance for each column are boldfaced. F1 is short for macro-F1.

Type	Model	Rest.					Laptop				
		ACC	**F1**	**Neg.**	**Neu.**	**Pos.**	**ACC**	**F1**	**Neg.**	**Neu.**	**Pos.**
Supervised methods	aLSTM [12]	76.83	66.48	66.89	44.67	87.88	68.07	64.82	59.64	53.39	81.42
	ATAE-LSTM [2]	76.61	65.41	66.19	43.34	86.71	67.40	59.41	55.27	42.15	80.81
	MemNet [10]	77.27	66.46	65.57	46.64	87.16	70.38	65.16	60.00	52.56	82.91
	RAM [4]	80.32	71.55	70.08	**55.28**	89.30	74.02	69.61	65.43	59.93	83.48
	MGAN [3]	81.25	71.94	–	–	–	75.39	72.47	–	–	–
	TNet [5]	80.41	70.56	71.20	51.34	89.14	**76.53**	71.93	**68.20**	60.14	**87.44**
	TRMN [21]	78.86	69.00	68.66	50.66	87.70	72.92	68.18	62.63	57.37	84.30
	IARM [22]	80.0	–	–	–	–	73.8	–	–	–	–
	HSCN [20]	77.80	70.20	–	–	–	76.10	**72.50**	–	–	–
Semi-supervised methods	PRET+MULT [19]	79.11	69.73	–	–	–	71.15	67.46	–	–	–
	ASVAET [16]	81.11	72.19	–	–	–	75.44	70.52	–	–	–
Our methods (NeaNet and its variants)	aLSTM*	80.27	69.50	69.79	50.77	87.94	73.82	69.39	64.94	58.90	84.32
	aLSTM*+NeSU	80.62	70.78	72.69	51.09	88.56	74.45	70.68	65.79	**61.20**	85.04
	SUCR-enabled aLSTM*	81.25	71.54	74.37	51.17	89.07	75.24	71.28	67.69	60.12	86.03
	SUSP-enabled aLSTM*	**82.05**	**72.58**	**74.75**	53.16	**89.83**	76.18	71.94	68.13	61.12	86.57
	NeaNet	**82.77**	**73.67**	**77.39**	**53.82**	**89.81**	**77.43**	**73.14**	**69.86**	**62.59**	**86.96**

best competing methods in the overall ACC, NeaNet substantially outperforms them by 1.18%–3.05% in Laptop and 2.64%–4.61% in Rest. The superiority of NeaNet is mainly due to the incorporation of the NeSU-driven SUCR and SUSP components that effectively leverage the discriminability of the NeSU to capture context-dependent sentiment information, which enables the LSTM networks to classify the sentiment of opinion targets more correctly. Particularly, as PRET+MULT is pretrained on document-level labeled sentiment data, its pretraining may introduce ambiguity for fine-grained sentiment task, leading to significantly less effective performance than NeaNet. ASVAET is also pretrained on unlabeled data, and generates generic representations only, which are much less expressive than the NeSU-enabled sentiment-aware representations.

Breakdown Performance. NeaNet obtains the best F1 performance in the negative class on both Rest. and Laptop, achieving 8.69% and 2.43% improvements over the best competing methods respectively. And NeaNet performs very competitive to the best results in positive and neutral classes. These results indicate that NeaNet well leverages unlabeled data to capture fine-grained sentiment features and achieves impressive improvements by using SUCR and SUSP.

4.4 Data Efficiency

This section is to answer whether the discriminability of NeSU enables NeaNet to achieve a more data-efficient learning. We evaluate the performance of NeaNet with randomly reduced training data, with RAM and TNet as the baselines.

The results are shown in Fig. 4. NeaNet performs significantly better than RAM and TNet in both ACC and macro-F1 with different amount of labeled training data on both Laptop and Rest. Particularly, even when NeaNet is trained using 50% less labeled data, it can obtain the ACC and/or macro-F1 performance that is comparable well to, or better than, RAM on both datasets. Similarly, NeaNet achieves comparable well performance to TNet even if 25% less training data is used in training NeaNet. This justifies that NeaNet can leverage the sentiment-aware property of NeSU to achieve substantially more effective exploitation of the small labeled data.

4.5 Ablation Study

NeaNet is compared with its four ablations as follows to investigate the contribution of its different components.

- aLSTM*: aLSTM* is a simple semi-supervised version of aLSTM by initializing with our pretrained VAE encoder.
- aLSTM*+NeSU: aLSTM*+NeSU is a simple NeSU-enabled aLSTM*, in which the NeSU-based sentiment value is added into the attention layer.
- SUCR-enabled aLSTM*: It is an enhanced aLSTM* with its plain LSTM replaced with SUCR. It is equivalent to NeaNet with SUSP removed.
- SUSP-enabled aLSTM*: It improves aLSTM* by replacing its LSTM with SUSP. It is a simplified NeaNet with SUCR removed.

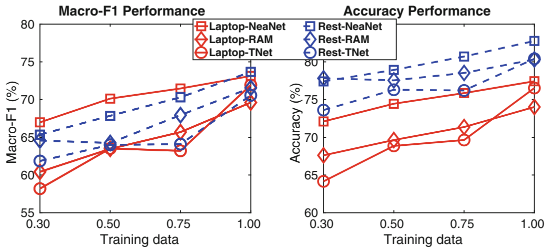


Fig. 4. Results with decreasing training data.

The results are given in the last group in Table 2. aLSTM* performs significantly better than aLSTM on all datasets, showing that the pretrained VAE can extract highly transferable features from unlabeled data. aLSTM*+NeSU, SUCR-enabled aLSTM* and SUSP-enabled aLSTM* outperform aLSTM* in all performance measures, which indicates that the discriminability of NeSU can enhance the downstream classifiers in various ways, e.g., to enhance the attention as in aLSTM*+NeSU or the memory architecture of LSTM as in SUCR/SUSP-enabled aLSTM*. SUCR/SUSP-enabled aLSTM* performs much better than aLSTM*+NeSU, indicating that SUSP and SUCR can exploit the power of NeSU more effectively; both of them underperform NeaNet, so both SUSP and SUCR are important to NeaNet. Particularly, SUSP-enabled aLSTM* performs consistently better than SUCR-enabled aLSTM*, revealing that, SUSP leverages the sentiment-aware property of NeSU to learn better representations than SUCR.

5 Conclusions

This paper introduces a novel semi-supervised approach to leverage large-scale unlabeled data for target-level sentiment classification on small labeled data. We discover for the first time that a few neuron units in encoding LSTM cells of the pretrained VAE demonstrate highly sentiment-discriminative capability. We further explore two effective ways to incorporate the most discriminative neural sentiment unit (NeSU) into attention networks to develop a novel LSTM-based target-level sentiment classifier. Empirical results show that our NeSU-enabled classifier substantially outperforms 13 state-of-the-art methods on two benchmark datasets and achieves significantly better data efficiency.

References

1. Xiaowen, D., Liu, B., Philip S.: A holistic lexicon-based approach to opinion mining. ACM (2008)
2. Wang, Y., Huang, M., Zhao, L.: Attention-based LSTM for aspect-level sentiment classification. In: EMNLP (2016)
3. Fan, F., Feng, Y., Zhao, D.: Multi-grained attention network for aspect-level sentiment classification. In: EMNLP (2018)

4. Chen, P., Sun, Z., Bing, L., Yang, W.: Recurrent attention network on memory for aspect sentiment analysis. In: EMNLP (2017)
5. Li, X., Bing, L., Lam, W., Shi, B.: Transformation networks for target-oriented sentiment classification. In: ACL (2018)
6. Wan, X.: Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In: EMNLP (2008)
7. Jiang, L.: Target-dependent twitter sentiment classification. In: ACL (2011)
8. Kiritchenko, S.: NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In: 2014 SemEval (2014)
9. Tang, D., Qin, B., Feng, X., Liu, T.: Target-dependent sentiment classification with long short term memory (2015)
10. Tang, D., Qin, B., Li, T.: Aspect level sentiment classification with deep memory network. In: EMNLP (2016)
11. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R.: Generating sentences from a continuous space (2015)
12. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: Effective attention modeling for aspect-level sentiment classification. In: COLING (2018)
13. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014)
14. Johnson, R., Zhang, T.: Semi-supervised convolutional neural networks for text categorization via region embedding (2015)
15. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by their probability distributions (1943)
16. Cheng, X., Xu, W., Wang, T., Chu, W.: Variational semi-supervised aspect-term sentiment analysis via transformer. In: CoNLL (2019)
17. Lei, Z., Yang, Y., Yang, M.: Sentiment lexicon enhanced attention-based LSTM for sentiment classification. In: AAAI (2018)
18. Fan, C., Qinghong, G., Du, J., Gui, L.: Convolution-based memory network for aspect-based sentiment analysis. In: ACM (2018)
19. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: Exploiting document knowledge for aspect-level sentiment classification. In: ACL (2018)
20. Lei, Z., Yang, Y., Yang, M., Zhao, W.: A human-like semantic cognition network for aspect-level sentiment classification. In: AAAI (2019)
21. Wang, S., Mazumder, S., Liu, B., Zhou, M.: Target-sensitive memory networks for aspect sentiment classification. In: ACL (2018)
22. Majumder, N., Poria, S.: IARM: inter-aspect relation modeling with memory networks in aspect-based sentiment analysis. In: EMNLP (2018)
23. Gururangan, S., Dang, T., Card, D., Smith, N.A.: Variational pretraining for semi-supervised text classification. In: ACL (2019)
24. Liu, M., Wen, M.: Semi-supervised learning with auxiliary evaluation component for large scale e-commerce text classification. In: ACL (2018)
25. Miyato, T., Dai, A.I., Goodfellow, I.: Adversarial training methods for semi-supervised text classification (2016)
26. Zhou, S., Chen, Q., Wang, X.: Fuzzy deep belief networks for semi-supervised sentiment classification. Neurocomputing **131**, 312–322 (2014)
27. Li, Y., Ye, J.: Learning adversarial networks for semi-supervised text classification via policy gradient. In: ACM (2018)
28. Bao, L., Lambert, P., Badia, T.: Attention and lexicon regularized LSTM for aspect-based sentiment analysis. In: ACL (2019)
29. Nguyen, T.H., Shirai, K.: Phrasernn: phrase recursive neural network for aspect-based sentiment analysis. In: EMNLP (2015)
30. Pontiki, M.: Semeval-2016 task 5: Aspect based sentiment analysis. In: 2016 SemEval(2016)