


Article

Multi-Task Learning for Sentiment Analysis with Hard-Sharing and Task Recognition Mechanisms

Jian Zhang ¹, Ke Yan ^{1,2,*}  and Yuchang Mo ³

¹ Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province, College of Information Engineering, China Jiliang University, Hangzhou 310018, China; p1903085237@cjl.u.edu.cn

² National University of Singapore, 4 Architecture Drive, Singapore 117566, Singapore

³ Fujian Province University Key Laboratory of Computational Science, School of Mathematical Sciences, Huaqiao University, Quanzhou 362021, China; myc@hqu.edu.cn

* Correspondence: keddiyan@gmail.com; Tel.: +65-9875-5205

Abstract: In the era of big data, multi-task learning has become one of the crucial technologies for sentiment analysis and classification. Most of the existing multi-task learning models for sentiment analysis are developed based on the soft-sharing mechanism that has less interference between different tasks than the hard-sharing mechanism. However, there are also fewer essential features that the model can extract with the soft-sharing method, resulting in unsatisfactory classification performance. In this paper, we propose a multi-task learning framework based on a hard-sharing mechanism for sentiment analysis in various fields. The hard-sharing mechanism is achieved by a shared layer to build the interrelationship among multiple tasks. Then, we design a task recognition mechanism to reduce the interference of the hard-shared feature space and also to enhance the correlation between multiple tasks. Experiments on two real-world sentiment classification datasets show that our approach achieves the best results and improves the classification accuracy over the existing methods significantly. The task recognition training process enables a unique representation of the features of different tasks in the shared feature space, providing a new solution reducing interference in the shared feature space for sentiment analysis.

Keywords: text classification; multi-task learning; hard-sharing mechanism; task recognition mechanism



Citation: Zhang, J.; Yan, K.; Mo, Y. Multi-Task Learning for Sentiment Analysis with Hard-Sharing and Task Recognition Mechanisms. *Information* **2021**, *12*, 207. <https://doi.org/10.3390/info12050207>

Academic Editor: Diego Reforgiato Recupero

Received: 19 March 2021
Accepted: 10 May 2021
Published: 12 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the fast development of e-commerce, the automated sentiment classification (ASC) method for reviews on various products is demanded in the field of nature language processing (NLP) [1]. ASC methods classify the reviews into positive/negative sentiment classes with satisfactory efficiency and accuracy [2]. More specifically, ASC intends to explore the in-depth attitudes and perceptions (such as positive, negative) from the text body associated with the user natural awareness.

Recently, many forms of neural networks (NN) have been proposed for ASC [3–5]. Inspired by the human behaviors that handle multiple tasks simultaneously, multi-task learning neural network (MTL-NN) is proposed, extending the NN with a more sophisticated internal structure. The MTL-NN is a hierarchical structure of NN performing sentiment analysis receiving data containing multiple tasks as input [1]. For example, an online shopping website contains review comments associated with various products, such as books, televisions, handphones, etc. Traditional single-task learning (STL) NN experiences difficulties analyzing text pieces mixing different product types. MTL-NN handles the entire text piece involving comments under different products. There are in general two main mechanisms for multi-task learning methods: (a) the soft-sharing mechanism that applies a task-specific layer to different tasks [6–8]; (b) the hard-sharing mechanism that

utilizes a powerful shared feature space to extract features for different tasks [9–11]. There are advantages and limitations for both sharing mechanisms.

With the continuous development of different versions of MTL, the soft-sharing mechanism has been widely adopted for ASC under different situations. However, there exist problems for the soft-sharing mechanisms, such as handling the interference between tasks and insufficient feature representations [12]. To address the above-mentioned issues, this paper proposes a sentiment analysis model that is based on the hard-sharing mechanism. A task recognition mechanism is proposed, which allows each task to obtain a unique representation in the hard-sharing feature space. The implemented model consists of three main steps. The first step consists of a lexicon encoder, which is used to encode the input data. It adds position and segment embedding to the word embedding. The second step contains a shared encoder, which is used to extract features from the data of several different tasks. These features form a shared feature space that provides supportive features for the subsequent private layers. The third step employs a private encoder, which consists of two layers: one is the task-specific layer for recognizing sentiment information, and the other is the task recognition layer.

The main contributions of our study can be summarized as follows:

- The proposed model addresses the issue of interference and generalization of the shared feature space during multi-task learning.
- The proposed model comprises three encoders, including a lexicon encoder, a shared encoder, and a private encoder, to improve the quality of extracted features.
- We propose a task recognition mechanism that makes the shared feature space have unique representation for different tasks.

2. Related Works

As one of the popular fields of natural language processing (NLP) [13,14], various sentiment classification methods were proposed in the recent years. For example, the Word2vec [15] technique, proposed by Google in 2013, significantly improves the traditional feature engineering methods for text classification. The Word2vec maps characters into low-dimensional vectors, representing the intrinsic connections between words [16,17]. The Word2vec accelerates the development of deep learning techniques in the field of sentiment classification

More recently, various deep learning algorithms were proposed for sentiment analysis, such as TextCNN [18], TextRNN [7], HAN [19], etc. These algorithms use different neural networks to process the text of different lengths. For example, convolutional neural networks [20,21] are used to extract features from sentences. Recurrent neural networks are used to extract features from paragraphs [22,23], and attention mechanisms are used to extract features from articles [19,24]. However, these algorithms cannot be directly applied to multi-task sentiment analysis.

The MTL approach allows the model to extract features from multiple tasks simultaneously. The MTL technique was firstly used in the field of computer vision [25,26]. Numerous experiments have demonstrated that MTL is better than single-task learning methods on multi-task sentiment analysis [27,28]. The latent correlations among similar tasks that can be extracted by MTL are potentially helpful in improving the classification results.

Based on the neural network structure, MTL can be divided into soft-sharing MTL and hard-sharing MTL [29]. The soft-sharing mechanism divides the features into shared features and private features, which reduces the interference between multiple tasks [30,31]. However, it requires learning separate features for each task as private features. These private features are not shared. Thus, the parameters are not used effectively [32]. The hard-sharing mechanism allows the shared layer to extract features from all tasks, which can be used by all tasks [6]. Multiple tasks interfere with each other in the shared layer, but the interference between tasks is exploited to improve generalizability. To reduce the interference, the hard-sharing mechanism provides a private layer for each task [33].

3. Methodology

The overall structure of the model is shown in Figure 1, where the lexicon encoder is used to encode the data for each task into a lexicon embedding. The shared encoder extracts the semantic features from the embedding. A shared feature space is formed by the semantic features extracted by the shared encoder. The private encoder consists of two parts: one is task-specific layers, which are used to learn semantic features related to the source of the review, and the other is the task recognition layer.

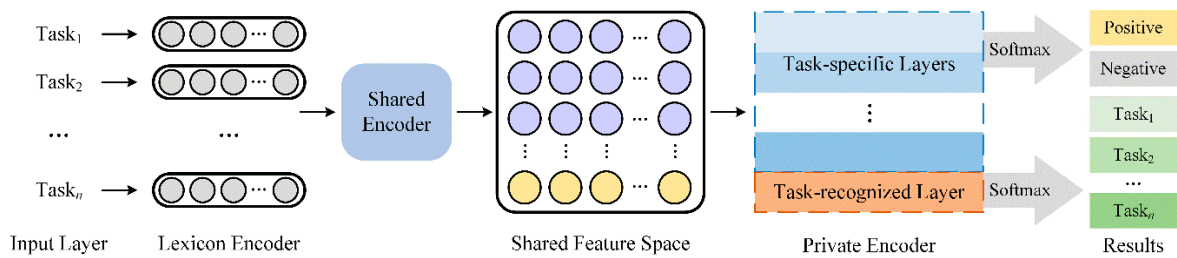


Figure 1. The overall framework of the proposed MTL-REC model.

3.1. The Lexicon Encoder

The lexicon encoder is a feature extraction encoder that addresses the issue of converting input text to word vectors. The input to the lexicon encoder can be a sentence or a paragraph. The output of the encoder is usually the representation of the sum of corresponding token, segment, and position embedded. The embedded position is calculated from the positions of input vectors, as shown in Equation (1).

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right), \tag{1}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right), \tag{2}$$

where pos is the position of the input vector; i is the dimension, and d_{model} is the dimension of the word vector. The lexicon encoder converts the input X into d_{model} dimension embeddings for the shared encoder learning (Section 3.2).

3.2. Shared Encoder

The shared encoder is used to extract the common sentence features in multiple tasks and places them into a shared feature space. To make the shared feature space contain richer semantic features, the pre-trained Bidirectional Encoder Representations from Transformers (BERT) model [34,35] is introduced into our shared encoder. The pre-trained BERT model consists of multiple Transformer Encoders, which can be used to encode sentences. The structure of the shared encoder is shown in Figure 2.

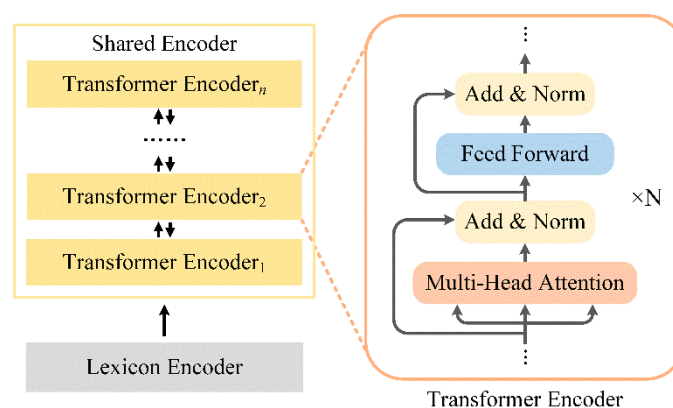


Figure 2. Structure of the shared encoder.

From Figure 2, the transformer encoder model is composed of a stack of $N = 6$ identical layers and specifically addresses the issue of learning long-term dependencies, which are composed of multi-head attention mechanisms and position-wise feed-forward networks. The two sub-layers are connected by residual connection [20] and layer normalization [21].

The multi-head attention allows the model to pay attention to the information in different locations. Multi-head attention is composed of multi-dimensional self-attention, which linearly projects the query keys and values h times. The self-attention consists of queries and keys of dimension d_k , and values of the dimension d_v . We compute the self-attention products with Equation (3):

$$Att(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (3)$$

where Q is the queries matrix; K is the keys matrix; V is the values matrix; and d_k is the dimension of the queries and keys. Finally, we utilize the softmax function to calculate the weight of every input token.

On each of the projected versions, the self-attention is computed in parallel. The outputs are concatenated. The final output can be calculated by projecting them again.

$$H(Q, K, V) = (h_1 \oplus h_2 \oplus \dots \oplus h_n)W^O, \quad (4)$$

$$h_i = Att(QW_i^Q, KW_i^K, VW_i^V), \quad (5)$$

where \oplus is the concatenation operator; h_i is the i -th attention representation of multi-head attention. The projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^O \in \mathbb{R}^{d_{\text{model}} \times d_k}$.

Position-wise feed-forward networks are composed by two linear transformations and a nonlinear activation function Relu:

$$FFN(x) = W_2 \cdot \text{Relu}(W_1 x + b_1) + b_2 \quad (6)$$

We utilize residual connections and layer normalizations to connect the input layer, multi-head attention mechanisms, and position-wise feed-forward networks:

$$LN(x) = \sigma(x + \mathcal{F}(x)), \quad (7)$$

where $\mathcal{F}(x)$ represents the output of sub-layers; σ represents layer normalization; $LN(x)$ represents the output of the layer normalization.

3.3. Private Encoder

The private encoder is composed of a task-specific layer and a task recognition layer. The task-specific layer is used to extract emotion features that are independent of tasks. Therefore, there are multiple multi-scale CNN layers, which are designed for different tasks. The task recognition layer is used to learn task-recognized features. The overall structure of the private encoder is shown in Figure 3.

From Figure 3, the multi-scale CNN is composed of multiple convolution layers. Each convolution layer is composed of multiple convolution kernels of different sizes that are used to extract text features of different scales in the shared feature space.

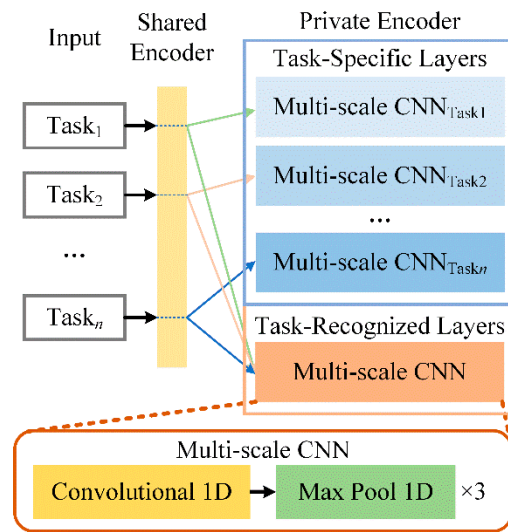


Figure 3. Structure of the private encoder.

Let $x_{i:i+j}$ refer to the concatenation of words $x_i, x_{i+1}, \dots, x_{i+j}$. A convolution operation is a convolution filter $w_h \in \mathbb{R}^{hk}$ sliding on a window of size h to generate new features. For example, convolution is calculated on the words $x_{i:i+h-1}$. A new feature can be generated by

$$c_i = f(w_h \cdot x_{i:i+h-1} + b), \quad (8)$$

where $b \in \mathbb{R}$ is a bias term; f is the ReLU activation function.

We apply the convolution filter to all possible word combinations $\{x_{1:h}, x_{2:h+1}, \dots, x_{n-h+1:n}\}$. A feature map can be generated by:

$$c_h = [c_1, c_2, \dots, c_{n-h+1}], \quad (9)$$

where $c_h \in \mathbb{R}^{n-h+1}$. We apply the max-pooling operation [3] to further process the feature c_h . The maximum value of c_h as a feature.

$$\hat{c}_h = \max\{c_h\}. \quad (10)$$

Multiple features \hat{c}_h of different length h are extracted by multiple convolution filters of different sizes, which represent token information of different lengths. The final features \hat{c} are concatenated by the multiple features \hat{c}_h extracted by convolution kernels of different sizes.

3.4. The Task Recognition Mechanism

Inspired by adversarial training [32], we propose a task recognition mechanism that uses the three encoders to learn the different features between each task while performing sentiment classification.

In the training process, for a text dataset containing N samples $\{x_i, y_i\}$, we utilize the cross-entropy function as the loss function. It is calculated that the cross-entropy of the true and the predicted distributions occurs on all the tasks. The model is optimized in the direction of minimizing the cross-entropy value.

$$L(\hat{y}, y) = - \sum_{i=1}^N \sum_{j=1}^C y_i^j \log(\hat{y}_i^j), \quad (11)$$

where y_i^j is the ground-truth label; \hat{y}_i^j is prediction probabilities, and C is the class number.

Task Discriminator. The task discriminator is used to map the shared representation of sentences into a probability distribution, estimating the probabilities of the original task for the encoded sentences.

During the task recognition training process, a separate multi-scale CNN layer is designed for each task. There are independent parameters in different multi-scale CNN layers. Therefore, the interference between different tasks can be relieved. Suppose that the input sample belongs to task k , the corresponding multi-scale CNN is $MCNN^{(k)}$. The output is:

$$\hat{y}^{(k)} = MCNN^{(k)}(x^{(k)}), \tag{12}$$

where $x^{(k)}$ is a sample of task k ; $\hat{y}^{(k)}$ is prediction probabilities of task k . For the data of multiple tasks, we calculate the weighted sum of the loss for each task.

$$L_{Task} = \sum_{k=1}^K \alpha_k L(\hat{y}^{(k)}, y^{(k)}), \tag{13}$$

where α_k is the weight for each task k . K is the number of tasks.

A task recognition training process is designed to learn different features from among tasks and influence the representation in the shared feature space by backpropagation. The schematic diagram of task recognition training is shown in Figure 4.

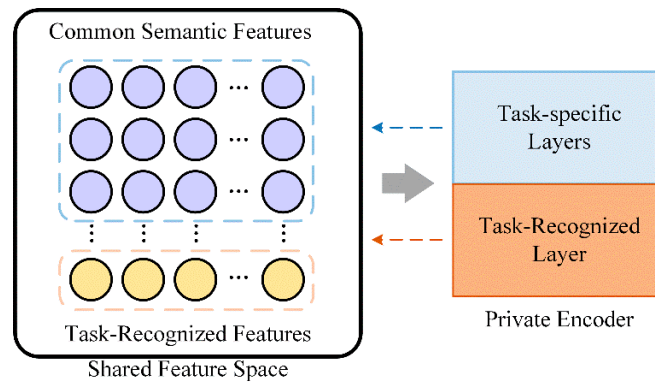


Figure 4. The task recognition training process.

Recognition Loss. Different with most existing multi-task learning algorithms, we add an extra recognition loss L_{rec} to add task-recognized features to shared feature space. The recognition loss is used to train a model to produce task-recognized features such that a classifier can reliably predict the task based on these features. The original loss of the task recognition training process is limited since it can only be used in binary situations. To overcome this, we extend it to multi-class form, which allows our model to be trained together with multiple tasks:

$$L_{rec} = -\frac{1}{K} \sum_{k=1}^K \sum_{i=1}^N \sum_{j=1}^C p_i^j \log(\hat{p}_i^j) \tag{14}$$

where K is the total number of tasks, N is the total number of samples, and C is the number of samples for task i . For each i , there are samples $j \in (1, C)$. p_i^j represents the predicted task that the sample j belongs to. Therefore, p_i^j is the task label, and \hat{p}_i^j is prediction probability of p_i^j . It is noted that the L_{rec} requires only the input sentence x and does not require the corresponding label y . The final loss function of the model can be written as:

$$L = L_{task} + \lambda L_{rec}, \tag{15}$$

where λ is a constant coefficient.

4. Experimental Process and Results

4.1. Dataset and Metrics

As shown in Table 1, the dataset that we employed in this experiment contains 16 different datasets from several popular review corpora, including books, electronics, DVD, kitchen, apparel, camera, health, music, toys, video, baby, magazines, software, sports, IMDB, and MR. The first 14 datasets are product reviews, which are collected by Blitzer et al. [33]. The remaining two datasets are movie reviews, which are from the IMDB datasets [34] and the MR datasets [18]. There are about 2000 reviews for each commodity, for a total of about 32,000 reviews. The goal is to classify a review as either positive or negative. All the datasets in each task are partitioned randomly into the training set, validation set, and test set with the proportion of 70%, 20%, and 10%, respectively. The detailed statistics about all the datasets are listed in Table 2.

Table 1. Instances of the testing dataset I.

Commodity Type	Example	Label
Books	this is a resource used by all nps i have talked to. great addition to your library.	1
	it was a mistake to buy it. only few pages were interestin	0
Electronics	great product but is only \$ 30 at iriver.com's stor	1
	i dont like this mouse, i brought, and never work, its useles	0
DVD	an awesome film with some suspense and raunchiness all rolled in to one	1
	i love pablo's act on comedy central. this one does n't even touch it	0
Kitchen	it is very light and worm. i love it. definitely worth the price!	1
	for the price, you get what you pay for. they are not the best quality	0
Apparel	recipient was very satisfied with this blanket as pb are his initials	1
	a red star !?!? i bet this wo n't sell well in eastern europe.	0
Camera	everything was excellent. the digital camera, the delivery. thank you a lot !!!!	1
	have had it for a few weeks and glad i brought it great procuc	0
Health	great tasting bar. nice and soft make it easy to eat	1
	it does n't get hot enough, nor does it stay hot for more than 10 min	0
Music	i just love lynch mixed with dooms production. it is what real is	1
	this cd isnt real good if you like compilations than get the ruff ryders c	0
Toys	these make meals a lot more fun for children... i know my son loves them	1
	fisher price is selling the same item for only \$ 33. \$ 139.99 has to be a mistake	0
Video	this is an excellent documentary of shangri-la and its elusive transcendental nature	1
	i love norm macdonald and this is the dumbest movie of all tim	0
Baby	great product—i heard from other mommies that this was the pump to get; i agree	1
	rent a hospital grade medalia pump. you wont be sorr	0
Magazines	the magazine was shipped in a timely manner, i would use this vendor again	1
	i still have not received this magazine, what is taking so long !!	0
Software	my husband is using the rosetta stone spanish program and loves it	1
	the "bad serial number" routine as the first reviewer.	0
Sports	excellent quality; much easier to put on than the cap i used before	1
	this pillow is too small and it is not comfortable at all	0
IMDB	this is a truly magnificent and heartwrenching film !!!	1
	argh! this film hurts my head. and not in a good way.	0
MR	it's a feel-good movie about which you can actually feel good.	1
	a decidedly mixed bag.	0

Table 2. Dataset I statistics.

Commodity Type	Training Set		Validation Set		Test Set		Total
	Positive	Negative	Positive	Negative	Positive	Negative	
Books	798	802	105	95	97	103	2000
Electronics	705	693	97	103	198	202	1998
DVD	802	798	95	105	102	98	2000
Kitchen	706	694	102	98	192	208	2000
Apparel	690	710	95	105	215	185	2000
Camera	706	692	99	100	194	206	1997
Health	812	788	98	102	90	110	2000
Music	698	702	103	97	199	201	2000
Toys	794	806	99	101	107	93	2000
Video	694	706	93	107	213	187	2000
Baby	800	700	103	97	97	103	1900
Magazines	682	688	101	99	217	183	1970
Software	788	727	102	98	110	90	1915
Sports	712	687	98	102	190	210	1999
IMDB	795	805	98	102	101	99	2000
MR	778	822	102	98	106	94	2000
Total	11,960	11,820	1590	1609	2428	2372	31,779

In addition, we collected four different types of commodity review datasets of daily necessities, literature, entertainment, and media from the raw data provided by Blitzer et al. [33] and formed dataset II. Each item in dataset II has more entries compared to dataset I. We also divided the training set, validation set, and test set for dataset II, and ensured that the number of positive and negative samples in each set did not differ much. Instances and statistics of dataset II are shown in Tables 3 and 4.

Table 3. Instances of the testing dataset II.

Commodity Type	Example	Label
Daily Necessities	great product—I heard from other mommies that this was the pump to get; i agree	1
	rent a hospital grade medalia pump. you wont be sorr	0
Literature	an excellent book for anyone that barbecues	1
	impossible to do so with no item received	0
Entertainment	thank you, i like this program and it does what i need it to do	1
	i would not buy it ! hard to use. my machine runs slower since the install.	0
Media	i received “the piano” promptly, and in pristine, excellent condition.	1
	if this is n’t worst dead album then in the dark is	0

Table 4. Dataset II statistics.

Commodity Type	Training Set		Validation Set		Test Set		Total
	Positive	Negative	Positive	Negative	Positive	Negative	
Daily Necessities	1609	1486	199	199	187	213	3893
Literature	2257	2305	308	292	420	380	5962
Entertainment	2285	2219	299	301	407	393	5904
Media	2978	3007	389	411	613	584	7982
Total	9129	9017	1195	1203	1627	1570	23,741

In the experiment, we use the same evaluation criteria for each commodity review data set and each method, which are accuracy and F1-score.

4.2. Compared with Other Sentiment Classification Methods

In order to verify the effectiveness of our proposed MTL-REC sentiment classification model, we select seven existing sentiment classification models for comparative study, including CNN [16], LSTM [35], bidirectional LSTM (Bi-LSTM) [36], LSTM with Attention (LSTM_Att) [17], MTL-CNN [7], MTL-GRU [37], MTL-ASP [12]. The initial hyper-parameter settings for all deep learning models include: number of hidden layers: 3; hidden layer size: 64; convolutional kernel sizes of the three hidden layers: 3, 4, and 5, respectively (only for CNN); optimizer: Ranger; learning rate: 0.2; dropout rate 0.5; epoch: 5; $\lambda = 0.5$. The source code of all models is available at: <http://www.github.com/zhang1546/Multi-Task-Learning-for-Sentiment-Analysis.git> (accessed on 12 May 2021).

The experimental results are shown in Tables 5 and 6.

Table 5. Performance of single-task model and multiple tasks on multiple tasks dataset I.

Task	Single Task					Multiple Tasks			
	CNN	LSTM	Bi-LSTM	LSTM_Att	Avg.	MTL-CNN	MTL-GRU	MTL-ASP	Proposed
Books	0.87	0.865	0.9	0.9	0.884	0.89	0.88	0.84	0.915
Electronics	0.825	0.84	0.848	0.852	0.841	0.862	0.842	0.868	0.885
DVD	0.8	0.835	0.87	0.855	0.840	0.85	0.82	0.855	0.875
Kitchen	0.848	0.878	0.85	0.855	0.858	0.86	0.872	0.862	0.865
Apparel	0.875	0.865	0.872	0.86	0.868	0.855	0.872	0.87	0.895
Camera	0.855	0.878	0.865	0.85	0.862	0.88	0.892	0.892	0.888
Health	0.845	0.855	0.865	0.83	0.849	0.885	0.875	0.882	0.865
Music	0.825	0.838	0.825	0.812	0.825	0.842	0.83	0.825	0.845
Toys	0.845	0.875	0.89	0.88	0.873	0.855	0.865	0.88	0.875
Video	0.872	0.88	0.882	0.875	0.877	0.878	0.885	0.845	0.91
Baby	0.885	0.875	0.875	0.855	0.873	0.89	0.9	0.882	0.865
Magazines	0.852	0.85	0.865	0.855	0.856	0.882	0.9	0.922	0.9
Software	0.89	0.905	0.885	0.885	0.891	0.905	0.895	0.872	0.91
Sports	0.858	0.858	0.85	0.84	0.852	0.875	0.862	0.857	0.908
IMDB	0.84	0.86	0.89	0.875	0.866	0.865	0.855	0.855	0.925
MR	0.73	0.74	0.715	0.755	0.735	0.72	0.7	0.767	0.79
AVG	0.845	0.856	0.859	0.852	0.853	0.862	0.859	0.861	0.882
STD	0.0373	0.0348	0.0416	0.0326	0.0347	0.0402	0.0471	0.0327	0.0321

Table 6. Performance of single-task model and multiple tasks on multiple tasks dataset II.

Task	Single Task					Multiple Tasks			
	CNN	LSTM	Bi-LSTM	LSTM_Att	Avg.	MTL-CNN	MTL-GRU	MTL-ASP	Proposed
Daily Necessities	0.850	0.850	0.865	0.852	0.854	0.855	0.848	0.865	0.878
Literature	0.860	0.834	0.845	0.831	0.843	0.851	0.829	0.850	0.865
Entertainment	0.870	0.861	0.851	0.878	0.865	0.874	0.869	0.860	0.898
Media	0.845	0.854	0.865	0.863	0.857	0.845	0.866	0.858	0.880
AVG	0.856	0.850	0.857	0.856	0.855	0.856	0.853	0.858	0.880
STD	0.0108	0.00991	0.00876	0.0171	0.00805	0.0108	0.0160	0.00540	0.0118

Table 5 shows the accuracy of 16 sentiment classification tasks. Table 6 shows the accuracy of four sentiment classification tasks. The column of Avg. shows the average accuracy of the previous four single models. The highest accuracy rates are bolded in Tables 5 and 6. From Table 5, we can see that multi-task learning models work better than single tasks in most tasks. From Table 6, we can see our proposed MTL-REC model outperforms all compared existing methods in all the cases. The classification accuracy improvements are visualized in Figure 5. In Figure 5, it is noted that the classification accuracy improvement with the proposed method over all compared methods is from 2% to 7%. The significant classification accuracy improvement

is mainly achieved by the hard-sharing mechanism and the task recognition training process. The task sharing layer reduces the interference between multiple tasks. Table 5 also shows that the CNN extracts text features similar to that of GRU and LSTM encoders and takes less time. In Table 6, the average accuracy of the multi-task learning model is almost the same as that of the single-task learning model. Table 7 shows a statistical test over the results shown in Table 5. The difference between the proposed method and each compared method is evaluated using the Wilcoxon signed-rank test. The *p*-values show that the proposed method is significantly different from the compared methods. Table 8 shows the overall time and memory used by different methods on dataset I and II. The proposed MTL-REC encoder improves the sentiment classification performance significantly, but requires more running time and memory. The time complexity of the sentiment analysis is usually not the main concern, since the feature extract part can always be performed offline.

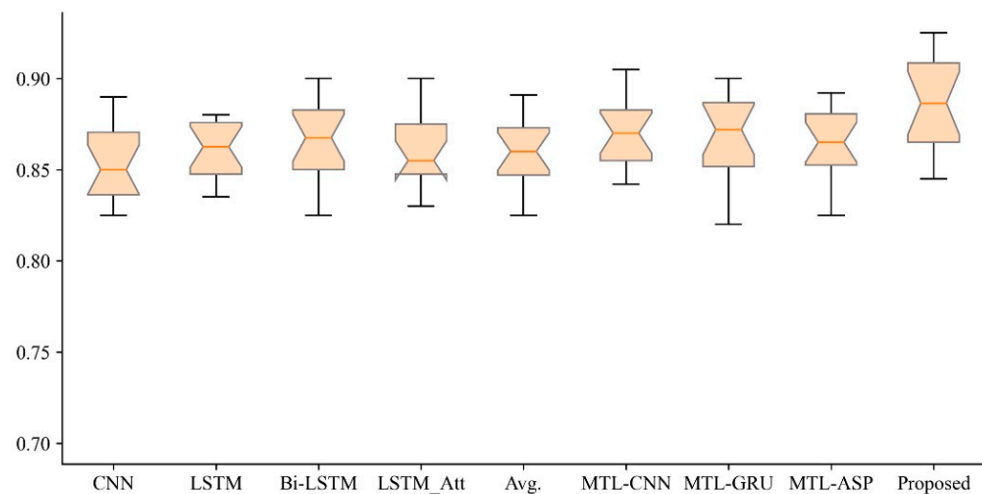


Figure 5. Visualization of the maximum, minimum, and averaged classification accuracy of all compared methods.

Table 7. The total time and memory used by different methods.

Statistical Methods	Levene’s Test		Wilcoxon Signed-Rank Test	
	<i>p</i> -Value	Evaluation	<i>p</i> -Value	Evaluation
Proposed-CNN	0.93	homogeneity of variance	1	significant difference
Proposed-LSTM	0.781	homogeneity of variance	0.998	significant difference
Proposed-Bi-LSTM	0.977	homogeneity of variance	0.999	significant difference
Proposed-LSTM_Att	0.717	homogeneity of variance	1	significant difference
Proposed-MTL-CNN	0.952	homogeneity of variance	0.997	significant difference
Proposed-MTL-GRU	0.723	homogeneity of variance	0.994	significant difference
Proposed-MTL-ASP	0.858	homogeneity of variance	0.991	significant difference

Table 8. The total time and memory used by different methods.

Task	Single Task				Multiple Tasks			
	CNN	LSTM	Bi-LSTM	LSTM_Att	MTL-CNN	MTL-GRU	MTL-ASP	Proposed
Time (s)	30.93	34.32	57.29	58.83	30.80	58.75	188.28	3282
Memory (MB)	577	647	647	647	577	647	649	1145

4.3. Model Self-Comparison

To demonstrate the effectiveness of the proposed method, a comparative experiment is conducted. Tables 9 and 10 reflect the fact that the BERT and task recognition training process are helpful in sentiment classification tasks.

Table 9. Performance improvement using BERT and task recognition training on multiple tasks dataset I.

Task	Without BERT		Without Task Recognition Mechanism		With BERT	
	Acc	F1	Acc	F1	Acc	F1
Books	0.915	0.915	0.91	0.91	0.915	0.912
Electronics	0.832	0.819	0.855	0.844	0.885	0.876
DVD	0.84	0.845	0.855	0.853	0.875	0.876
Kitchen	0.85	0.84	0.852	0.844	0.865	0.856
Apparel	0.865	0.87	0.892	0.9	0.895	0.902
Camera	0.878	0.873	0.88	0.881	0.888	0.888
Health	0.875	0.859	0.875	0.857	0.865	0.846
Music	0.832	0.835	0.862	0.859	0.845	0.845
Toys	0.875	0.886	0.87	0.883	0.875	0.886
Video	0.888	0.894	0.905	0.911	0.91	0.914
Baby	0.9	0.895	0.855	0.854	0.865	0.862
Magazines	0.878	0.881	0.878	0.887	0.9	0.91
Software	0.915	0.922	0.915	0.922	0.91	0.916
Sports	0.872	0.862	0.885	0.875	0.908	0.901
IMDB	0.91	0.91	0.91	0.913	0.925	0.925
MR	0.755	0.749	0.81	0.812	0.79	0.788
AVG	0.867	0.866	0.876	0.875	0.882	0.881
STD	0.0390	0.0417	0.0268	0.0300	0.0321	0.0346

Table 10. Performance improvement using BERT and task recognition training on multiple tasks dataset II.

Task	Without BERT		Without Task Recognition Network		With BERT	
	Acc	F1	Acc	F1	Acc	F1
Daily Necessities	0.85	0.84	0.852	0.841	0.878	0.869
Literature	0.864	0.867	0.869	0.875	0.865	0.871
Entertainment	0.854	0.854	0.884	0.884	0.898	0.899
Media	0.852	0.855	0.882	0.884	0.88	0.882
AVG	0.755	0.749	0.81	0.812	0.882	0.881
STD	0.00539	0.00957	0.0128	0.0177	0.0118	0.0119

In Tables 9 and 10, the highest accuracy rates and F1 scores are highlighted using bold font. According to Tables 9 and 10, the sentiment classification performance is further improved with BERT and the proposed task recognition mechanism.

5. Conclusions

In this paper, we propose a multi-task learning framework for sentiment classification with a novel task recognition mechanism. We introduce the pre-trained BERT as our shared encoder to further improve the performance of the shared encoder. In addition, we propose a task recognition training process, which enhances the shared feature space to obtain more task-recognized features. We designed a series of experiments to validate our proposed method. The experimental results show that the sentiment classification results of our proposed model are superior to existing state-of-art methods. Both semantic features and task-recognized features are extracted, enhancing the overall classification performance.

It is noted that we introduce the pre-trained BERT model, which reduces the efficiency of the algorithm and leads to longer computation times. The proposed method shows a significant improvement on the accuracy of sentiment classification.

As one of the future works, we will improve on the shared encoder to reduce the time complexity of the proposed multi-task learning algorithm. In addition, more challenging datasets, such as unbalanced, noisy datasets, and datasets in different languages, will be tested on the proposed method.

Author Contributions: Conceptualization, K.Y.; methodology, K.Y.; software, J.Z.; validation, K.Y.; formal analysis, K.Y.; investigation, Y.M.; resources, Y.M.; data curation, Y.M.; writing—original draft preparation, K.Y.; writing—review and editing, K.Y.; visualization, J.Z.; supervision, K.Y. and Y.M.; project administration, K.Y. and Y.M.; funding acquisition, K.Y. and Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LY19F020016 (K.Y.), in part by the National Natural Science Foundation of China under Grant 61972156, and Program for Innovative Research Team in Science and Technology in Fujian Province University (Y.M.).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code and data used in this study is freely available at an open-source version control website: <http://www.github.com/zhang1546/Multi-Task-Learning-for-Sentiment-Analysis.git> (accessed on 12 May 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Gómez-Adorno, H.; Fuentes-Alba, R.; Markov, I.; Sidorov, G.; Gelbukh, A. A convolutional neural network approach for gender and language variety identification. *J. Intell. Fuzzy Syst.* **2019**, *36*, 4845–4855. [[CrossRef](#)]
- Dejun, Z.; Mingbo, H.; Lu, Z.; Fei, H.; Fazhi, H.; Zhigang, T.; Yafeng, R. Attention Pooling-Based Bidirectional Gated Recurrent Units Model for Sentimental Classification. *Int. J. Comput. Intell. Syst.* **2019**, *12*, 723–732.
- Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* **2011**, *12*, 2493–2537.
- Misra, I.; Shrivastava, A.; Gupta, A.; Hebert, M. Cross-Stitch Networks for Multi-task Learning. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3994–4003.
- Liu, P.; Qiu, X.; Huang, X. Recurrent Neural Network for Text Classification with Multi-Task Learning. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; AAAI Press: Palo Alto, CA, USA; pp. 2873–2879.
- Ruder, S.; Bingel, J.; Augenstein, I.; Søgaard, A. Latent Multi-Task Architecture Learning. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4822–4829. [[CrossRef](#)]
- Collobert, R.; Weston, J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In Proceedings of the 25th international conference on Machine learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
- Subramanian, S.; Trischler, A.; Bengio, Y.; Pal, C.J. Learning General Purpose Distributed Sentence Representations via Large Scale Multi-Task Learning. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
- Liu, X.; He, P.; Chen, W.; Gao, J. Multi-Task Deep Neural Networks for Natural Language Understanding. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4487–4496.
- Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
- Bing, L. *Sentiment Analysis and Opinion Mining*; Morgan & Claypool: San Rafael, CA, USA, 2012.
- Liu, P.; Qiu, X.; Huang, X. Adversarial Multi-task Learning for Text Classification. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2017; pp. 1–10.
- Sun, S.; Luo, C.; Chen, J. A review of natural language processing techniques for opinion mining systems. *Inf. Fusion* **2017**, *36*, 10–25. [[CrossRef](#)]
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed Representations of Words and Phrases and Their Compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Lake Tahoe, CA, USA, 5–10 December 2013; Curran Associates Inc.: Red Hook, NY, USA, 2013; Volume 2, pp. 3111–3119.
- Nyberg, K.; Raiko, T.; Tiinanen, T.; Hyvönen, E. Document Classification Utilising Ontologies and Relations between Documents. In Proceedings of the Eighth Workshop on Mining and Learning with Graphs, Washington, DC, USA, 5 August 2010; Association for Computing Machinery: New York, NY, USA, 2010; pp. 86–93.

16. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2014; pp. 1746–1751.
17. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
18. Pang, B.; Lee, L. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, MI, USA, 25–30 June 2005; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 115–124.
19. Yanmei, L.; Yuda, C. Research on Chinese Micro-Blog Sentiment Analysis Based on Deep Learning. *2015 8th Int. Symp. Comput. Intell. Des.* **2015**, *1*, 358–361. [[CrossRef](#)]
20. Graves, A.; Jaitly, N.; Mohamed, A.-R. Hybrid speech recognition with Deep Bidirectional LSTM. In Proceedings of the 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, 8–13 December 2013; pp. 273–278.
21. Wen, S.; Wei, H.; Yang, Y.; Guo, Z.; Zeng, Z.; Huang, T.; Chen, Y. Memristive LSTM Network for Sentiment Analysis. *IEEE Trans. Syst. Man, Cybern. Syst.* **2021**, *51*, 1794–1804. [[CrossRef](#)]
22. Zhang, S.; Xu, X.; Pang, Y.; Han, J. Multi-layer Attention Based CNN for Target-Dependent Sentiment Classification. *Neural Process. Lett.* **2020**, *51*, 2089–2103. [[CrossRef](#)]
23. Caruana, R. Multitask Learning. *Mach. Learn.* **1997**, *28*, 41–75. [[CrossRef](#)]
24. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V. Domain-Adversarial Training of Neural Networks. *J. Mach. Learn. Res.* **2016**, *17*, 1–35.
25. Zhang, Z.; Luo, P.; Loy, C.C.; Tang, X. Facial Landmark Detection by Deep Multi-Task Learning. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 94–108.
26. Daumé, H. Bayesian Multitask Learning with Latent Hierarchies. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, Canada, 18–21 June 2009; AUAI Press: Arlington, VA, USA, 2009; pp. 135–142.
27. Sun, T.; Shao, Y.; Li, X.; Liu, P.; Yan, H.; Qiu, X.; Huang, X. Learning Sparse Sharing Architectures for Multiple Tasks. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Association for the Advancement of Artificial Intelligence (AAAI): Palo Alto, CA, USA, 2020; Volume 34, pp. 8936–8943.
28. Liu, P.; Qiu, X.; Huang, X. Deep Multi-Task Learning with Shared Memory for Text Classification. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2016; pp. 118–127.
29. Hessel, M.; Soyer, H.; Espeholt, L.; Czarnecki, W.; Schmitt, S.; Van Hasselt, H. Multi-Task Deep Reinforcement Learning with PopArt. In Proceedings of the 2019 AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Association for the Advancement of Artificial Intelligence (AAAI): Palo Alto, CA, USA, 2019; Volume 33, pp. 3796–3803.
30. Liu, X.; Gao, J.; He, X.; Deng, L.; Duh, K.; Wang, Y.-Y. Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May–5 June 2015; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2015; pp. 912–921.
31. Li, J.; Monroe, W.; Shi, T.; Jean, S.; Ritter, A.; Jurafsky, D. Adversarial Learning for Neural Dialogue Generation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2017; pp. 2157–2169.
32. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Bangkok, Thailand, 18–22 November 2010; MIT Press: Cambridge, MA, USA, 2014; Volume 2, pp. 2672–2680.
33. Blitzer, J.; Dredze, M.; Pereira, F. Biographies, Bollywood, Boom-Boxes and Blenders: Domain Adaptation for Sentiment Classification. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, 25–27 June 2007; Association for Computational Linguistics: Stroudsburg, PA, USA, 2007; pp. 440–447.
34. Maas, A.L.; Daly, R.E.; Pham, P.T.; Huang, D.; Ng, A.Y.; Potts, C. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; pp. 142–150.
35. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
36. Graves, A.; Mohamed, A.-R.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2013; pp. 6645–6649.
37. Lu, G.; Gan, J.; Yin, J.; Luo, Z.; Li, B.; Zhao, X. Multi-task learning using a hybrid representation for text classification. *Neural Comput. Appl.* **2020**, *32*, 6467–6480. [[CrossRef](#)]