International Journal of Machine Learning and Cybernetics (2022) 13:2711–2724
https://doi.org/10.1007/s13042-022-01556-0

ORIGINAL ARTICLE

Learning to share by masking the non-shared for multi-domain sentiment classification

Jianhua Yuan[1] · Yanyan Zhao[1] · Bing Qin[1,2]

Received: 31 July 2021 / Accepted: 28 March 2022 / Published online: 7 May 2022

Abstract
Multi-domain sentiment classification deals with the scenario where labeled data exists for multiple domains but is insufficient for training effective sentiment classifiers that work across domains. Thus, fully exploiting sentiment knowledge shared across domains is crucial for real-world applications. While many existing works try to extract domain-invariant features in high-dimensional space, such models fail to explicitly distinguish between shared and private features at the text level, which to some extent lacks interpretability. Based on the assumption that removing domain-related tokens from texts would help improve their domain invariance, we instead first transform original sentences to be *domain-agnostic*. To this end, we propose the BERTMasker model which explicitly masks domain-related words from texts, learns domain-invariant sentiment features from these domain-agnostic texts and uses those masked words to form domain-aware sentence representations. Empirical experiments on the benchmark multiple domain sentiment classification datasets demonstrate the effectiveness of our proposed model, which improves the accuracy on multi-domain and cross-domain settings by 1.91% and 3.31% respectively. Further analysis on masking proves that removing those domain-related and sentiment irrelevant tokens decreases texts' domain separability, resulting in the performance degradation of a BERT-based domain classifier by over 12%.

Keywords Natural language processing · Sentiment analysis · Cross domain · Masking

1 Introduction

Sentiment classification [17, 22, 32] is one of the key tasks in Natural Language Processing. The recent success of sentiment classification relies heavily on deep neural networks trained with a large number of carefully annotated data. However, as the diversity of domains leads to the discrepancy of sentiment features, models trained on existing domains may not perform ideally on the domain of interest. Meanwhile, as not all domains have adequate labeled data, it is necessary to leverage existing annotations from multiple domains. For instance, in both DVD and Video domains, *picture* and *animation* can be opinion targets and *thrilling* and *romantic* are frequent polarity words. Exploiting such sharedness would help improve both in-domain and out-of-domain sentiment classification results.

In this work, we focus on the task of multi-domain sentiment classification (MDSC) where we need to make full use of limited annotated data and large unlabeled data from each domain to train a classifier that achieves the best average performance on all domains. There exist two major lines of work attempting to tackle this challenge. One line is to exploit the shared-private framework [1, 28], where domain-agnostic features are captured by the networks shared across all domains and domain-specific representations by the feature extractor of each domain. [4, 19] applied domain adversaries to shared features for better learning of domain-invariant representations. The other major line of work [2, 20, 33] implicitly utilized such share-private ideas where they first learned domain-specific query vectors (or domain embeddings) and then used these to compose domain-aware representation by attending features from shared sentence

✉ Bing Qin
qinb@ir.hit.edu.cn

Jianhua Yuan
jhyuan@ir.hit.edu.cn

Yanyan Zhao
yyzhao@ir.hit.edu.cn

[1] Faculty of Computing, Harbin Institute of Technology, Harbin 150001, China

[2] Pengcheng Lab, Shenzhen 518066, China

Example 1 (Domain: Sports)

this is a great **[helmet]** . my **[daughter]** has been very happy with it and loves to **[wear]** it

Example 2 (Domain: Books)

this **[cookbook]** is the best in my **[collection]** - and i have a lot ! ! the **[instructions]** are clear and the **[pictures]** are great

Fig. 1 Two examples from *Sports* and *Books* domains that illustrate our motivation for transforming sentences to be *domain-invarint* by masking domain-related words in square brackets

encoder. So far, these two major methods have not been effectively combined.

While shared-private models learn domain-agnostic features in vector space, the discrimination between shared and private features cannot be directly interpreted to humans at the text-level. Therefore, we propose to distinguish domain-related and domain-agnostic tokens before further feature extractions, based on the intuition that removing domain-related words from texts would help improve their domain-invariance. Given two sentences from *Sports* and *Book* domains respectively in Fig. 1, after removing domain-related words like *helmet* from *Sports* domain and *cookbook* from *Book* domain, these sentences become more domain-agnostic. Meanwhile, the most salient sentiment-related semantics are mostly preserved in the remaining texts. In this way, it would be possible to tell what features are domain-related and what features are shared by all domains to some extent.

To combine the advantage of both paradigms in multi-domain sentiment analysis, a model should employ the shared-private framework, where the shared part learns domain-agnostic sentiment features and the private part captures a domain-aware sentiment representation based on the shared feature extractors (contrary to using separate extractors for each domain in [19]). To learn good shared sentiment features with better interpretability at the text level, a model should be capable of discriminating between domain-related and domain agnostic tokens at first. To this end, we propose the BERTMasker model. The BERTMasker model learns to first select domain-related tokens from texts, then masks those tokens from the original text and acquires domain-agnostic sentiment features for the shared part. As the masked tokens are domain-related, they are appropriate for learning domain-aware sentiment representations of texts from different domains. We incorporate this advantage into the private part of BERTMasker. Since simple models are not adequate for learning good sentiment features from fractional texts, we turn to BERT [5] for text encoding as it shares a similar input format during its pre-training phase of Masked Language Model (MLM). Motivated by previous work [26] utilizing Next Sentence Prediction task in BERT, we also expect inputting texts with [MASK] at both training and inference time would boost the performance in multi-domain sentiment analysis tasks. Though we have no accurate prior knowledge of what domain-related tokens are, the BERTMasker takes a detour of learning domain-related tokens as we have some knowledge of what domain-related tokens should not be for our sentiment classification task. In other words, tokens from general sentiment lexicons and commonly used stopwords are domain-agnostic. We enhance this prior knowledge as constraints to our model and train a domain classifier to guide more accurate learning of domain-related tokens. Those tokens play important roles in learning both shared and private sentiment features.

Our contributions can be summarized as follows:

- We propose a novel model named BERTMasker to better learn shared representation across domains by masking domain-related tokens from texts.
- Our model combines both shared-private framework and domain-aware feature learning, where the token masking network in the shared part learns domain-invariant text transformation and in the private part aggregates domain-aware sentiment features.
- Evaluation results on benchmark multi-domain sentiment classification datasets demonstrate the superiority of our proposed model. Further analyses on masked tokens and remaining texts prove the plausibility and effectiveness of the token masking mechanism.

2 Related work

Our work uses a BERT-based model for multi-domain sentiment classification. We describe related work from these two perspectives. Since our model learns to mask domain-informative words, we also discuss relate work in domain words extraction for sentiment analysis.

2.1 Multi-domain sentiment classification

The task of multi-domain sentiment classification [16] aims at training models that leverage data from multiple domains to improve the overall classification performance on all domains. Currently, there exist mainly two lines of related methods. One line of methods [1, 4, 19, 20] is to exploit shared-private framework, where domain-agnostic features are usually captured with adversarial training or gradient reversal layer [6] at the shared part. Meanwhile, domain-specific representations are learned by feature extractors of each domain. Further, [3, 7] apply mixture-of-expert [13] approach to explicitly capture knowledge shared among similar domains. The other line of methods [2, 33] is to learn domain representations through domain classification and use these as queries to acquire domain-sensitive representations of input texts.

Our proposed BERTMasker combines the power of both paradigms. It employs a shared-private framework. It first learns to select domain-related words. Then, our model obtains shared sentiment features by exploiting texts without those words and uses these selected words to obtain domain-aware sentiment representations.

2.2 BERT-based models in sentiment analysis

BERT is one of the key techniques in the recent advances of contextualized representation learning [5, 8, 21, 23]. Its success relies on two pre-training tasks, namely Masked Language Model (MLM) and Next Sentence Prediction (NSP). Currently, there are mainly two ways to utilize BERT for downstream tasks. One is fine-tuning for each end task. For instance, to make the input format consistent with that of NSP, [26] constructs auxiliary sentences for aspect-based sentiment classification in four ways. And the other is injecting task-specific knowledge [15, 27] using new pre-training tasks. Such injections are usually done along with MLM where other objectives like POS tag, sentiment polarity [15], and sentiment targets [27] are introduced.

Our work is partially motivated by [26], as we both transform inputs to have the same format as one of the pre-training tasks of BERT (they use NSP while we use MLM instead). As MLM aims at predicting masked words based on context from both left and right, the BERT model can recover the semantic of the current word being masked. In other words, the BERT model could retain most of the features of the sentences while a small portion of its constituent tokens being masked. We make use of this advantage and design a model that could automatically learn to mask some (domain-related) words. In this way, a sentence could be transformed to be domain-invariant while still retaining its most salient sentiment features.

2.3 Domain words extraction

Domain words are usually referred to as domain-dependent sentiment words and target words in texts that are closely related to sentiment. Extracting those sentiment and target words is crucial for opinion mining. [9] proposed a dictionary-based method to extract sentiment words and used association-rules to identify target words. [24] introduced a semi-supervised double-propagation method to extract sentiment words and target words using syntactic rules and manually collected seed words. [18] utilized an RNN-based sequence labeling model to identify sentiment expressions in a supervised manner. [12] leveraged an LSTM-based model to extract target words. Similar to [9], we use a manual collected sentiment lexicon. However, in

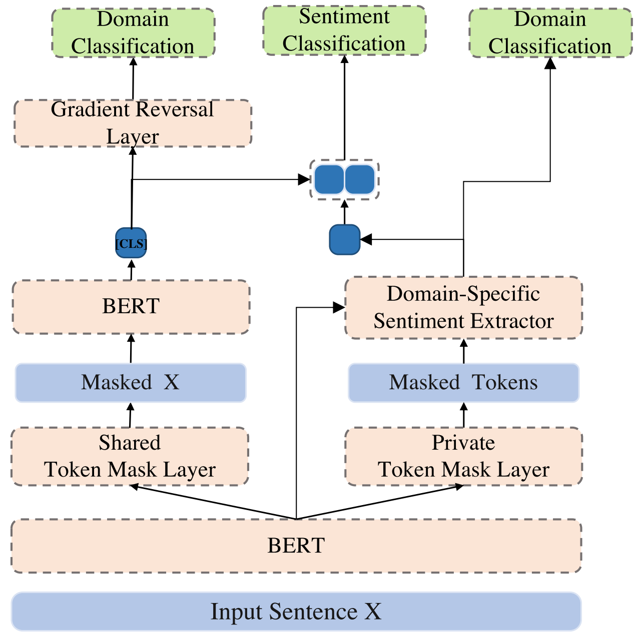


Fig. 2 The overall architecture of BERTMasker

our work, domain-informative tokens are not extracted by sequence labeling systems or syntactic rule-based methods. Instead, domain-informative tokens are selected according to whether they contribute to the identification of their corresponding domains. Besides, those domain words are jointly constrained by the sentiment classification task.

3 Model

3.1 Overview

An overview of our model is shown in Fig. 2. Basically, our model adopts the popular adversarial shared-private framework, where the shared part (the left part in Fig. 2) is utilized for extracting domain-invariant features and the private part (the right part in Fig. 2) for learning domain-specific features. For a given sentence, BERTMasker first encodes representations of each word in its context. Then it uses token masking networks (see Fig. 3) to select domain-related tokens based on these features for shared and private parts respectively. In the shared part, each domain-related token is replaced by a [MASK] symbol in the original text, and a more domain-invariant text is obtained. After that, BERTMasker feeds the masked texts into BERT again and learns domain-invariant sentiment features with domain-adversarial training. In the private part, domain-related tokens are utilized to learn domain-aware sentiment representations with attention mechanism (see Fig. 4). Finally, the concatenations of shared and private features are used

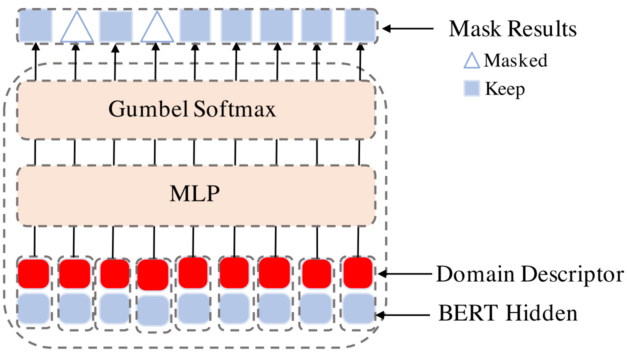


Fig. 3 Token masking network

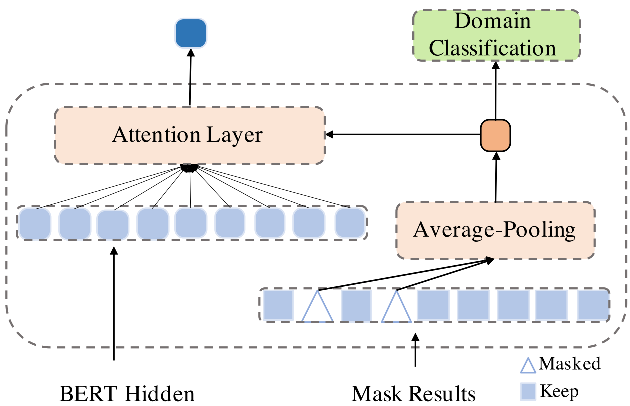


Fig. 4 Domain-specific sentiment feature extractor

for sentiment prediction. In the following, we introduce key components of our model in detail.

3.2 Sentence modeling with BERT

Given an input sequence $X = \{x_1, x_2, \dots, x_N\}$, we first transform it into the required format of BERT model as $X = \{[\text{CLS}], x_1, x_2, \dots, x_N, [\text{SEP}]\}$. Then we get the contextualized representation h_i of token x_i from the BERT encoder:

$$\begin{aligned}&\{h_{[\text{CLS}]}, h_1, h_2, \dots, h_N, h_{[\text{SEP}]}\}\\&\quad = \text{BERT}([\text{CLS}], x_1, x_2, \dots, x_N, [\text{SEP}])\end{aligned} \tag{1}$$

where N is the number of tokens in the input sequence.

The MLM task of BERT enables it to process sequences whose tokens are partially replaced by [MASK] symbol. We exploit such intrinsic advantage of BERT to facilitate our idea of modeling text after the removal of domain-related tokens. Suppose K tokens in the given sequence are selected and we have the masked text $\hat{X} = \{[\text{CLS}], x_1, x_2, \dots, [\text{MASK}]_1, \dots, [\text{MASK}]_K, \dots, x_N, [\text{SEP}]\}$. Then we could model the new text with Equation 1 and obtain $H^{masked} = \{h_{[\text{CLS}]}, h_1, h_2, \dots, h_{[\text{MASK}]_1}, \dots, h_{[\text{MASK}]_K}, \dots, h_N, h_{[\text{SEP}]}\}$.

Following previous methods using BERT, we can choose the hidden feature $h_{[\text{CLS}]}$ of token [CLS] as the sequence representation.

3.3 Token masking networks

It is intuitive that if we remove some domain-related tokens from a text, the remaining part should be more domain-agnostic than the original one. Motivated by this, we design the token masking networks (TMN) to automatically discriminate whether a token is domain-specific. Here, we describe how TMN selects domain-related words and generate masked results for the shared and private parts respectively.

3.3.1 Shared part

For a token x_i, TMN decides the masking result by measuring its relatedness to the domain of its corresponding sentence. Following [20], we also introduce domain descriptors $D = \{d_1, d_2, \dots, d_j, \dots, d_{|D|}\}$ for each domain, where $|D|$ is the number of domains involved in training and test. A domain descriptor d_j is an L dimensional vector that encodes the most representative characteristics of the jth domain. It is randomly initialized and is jointly trained with other networks using gradient descents. As domain labels are available at both training and test time, we can leverage those domain descriptors to help decide whether a token is highly correlated with a specific domain. For each token x_i, we combine its contextualized representation h_i and its domain descriptor d_j as $z_i = h_i \oplus d_j$, where $\oplus$ represents vector concatenation. We use simple feed-forward neural networks with *tanh* non-linearity for measuring relatedness π_i between a token and the domain of its text. Based on these relatedness scores, we can infer whether a token is domain-related and further remove those domain-specific ones from the original text. While we expect a discrete decision of mask, simply applying *argmax* operation on π_i may break the gradients and the model can not be end-to-end trained. To enable end-to-end training and generate discrete decisions of masks, we apply GumbelSoftmax [14] instead of softmax. This is achieved as follows:

$$\pi_i = W_{tm_2} tanh(W_{tm_1}(h_i \oplus d_j) + b_{tm_1}) + b_{tm_2} \tag{2}$$

$$p_i = \frac{(G_i + log(\pi_i))/\tau}{\sum_{l=1}^{2}(G_l + log(\pi_l))/\tau} \tag{3}$$

where W_{tm_1}, W_{tm_2}, b_{tm_1} and b_{tm_2} are weights and bias terms for measuring similarities respectively. $\oplus$ is the operation of vector concatenation. And $G_i \sim Gumbel(0, 1)$ are i.i.d.

samples drawn from the standard Gumbel distribution. τ is the temperature parameter that controls how closely the new samples approximate discrete, one-hot vectors. As $\tau \rightarrow 0$, the softmax computation smoothly approaches the argmax, and the sample vectors approach one-hot; as $\tau \rightarrow \infty$, the sample vectors become uniform. $p_i = 0$ means that a token is domain-invariant and $p_i = 1$ means that a token is domain-related.

We aggregate the masking result of each token in a sentence and denote it as $P_{shared} = \{p_1, p_2, \dots, p_i, \dots, p_N\}$.

3.3.2 Private part

Instead of only using the domain descriptor d_j of the current text, we adopt a mixture of domain descriptors for each input sequence in the private part. This modification is designed to better capture domain-related words for each sentence if it shares similarities with sentences from other domains. In our preliminary experiments, it consistently works better than only using the original domain descriptor. We treat $h_{[CLS]}$ as the current sentence representation and measure its relatedness to the ith domain using a simple feed-forward attention network as follows:

$$z_j = d_i \oplus h_{[CLS]} \tag{4}$$

$$s_{ij} = W_{tp_2} tanh(W_{tp_1} z_j + b_{tp_1}) + b_{tp_2} \tag{5}$$

$$a_{ij} = \frac{e^{s_{ij}}}{\sum_{m=1}^{|D|} e^{s_{mj}}} \tag{6}$$

where d_i is the ith domain descriptor and $|D|$ is the number of domains.

Then, the aggregated mixture-of-descriptors is obtained:

$$\hat{d}_j = \sum_{1}^{|D|} a_{ij} * d_i \tag{7}$$

We follow similar steps of Eqs. 2–3 except that d_j in Eq. 2 is replaced with $\hat{d}_j$. We denote the masking result of private part as $P_{private}$.

3.3.3 Sentiment knowledge-enhanced masking constraints

The masking process is directly affected by domain descriptors and indirectly affected by adversarial domain classification and sentiment classification in the latter part. While the above mentioned token masking networks generate good results for final sentiment classification, preliminary results show that the masked texts are less interpretable for humans due to that many irrelevant tokens are mis-classified as domain-related. This may owe to the existence of non-robust features [11] that can be easily captured in these multi-domain datasets.

However, due to the diversity of domains in multi-domain sentiment classification, we are unlikely to have prior knowledge of whether tokens are domain-related. Luckily, for sentiment analysis, we know that words from sentiment lexicons contain general sentiment features that are not domain-specific. Similarly, common stop words are not domain-related. With these heuristics, we take a detour to calibrate the masking results. Instead of pointing out which tokens are domain-related, we explicitly ignore masking decisions on tokens that are not domain-specific, namely tokens in manually annotated sentiment lexicons and stopword lexicons. Furthermore, we add common negation and intensifier words into the constraints. In our preliminary results, these sentiment knowledge-enhanced masking constraints reduce the masking rate from 30% to less than 15% on average by preventing those general tokens from affecting the masking process.

It is intuitive to use the same masking networks for both shared and private parts. However, these two masking networks do have different emphases. In the shared part, its goal is to identify tokens that are not domain-general. In the private part, it focuses on picking domain-discriminative tokens by leveraging a mixture of domain descriptors. Tokens from similar domains are also implicitly chosen in the private part. Besides, using different token masking networks allows us to control the strength of domain distinction by different coefficients of domain classification. Furthermore, in our preliminary experiments, using different networks works slightly better than using the same network.

3.4 Domain-invariant sentiment feature extraction

After acquiring the masking result P_{shared} from the token masking network in the shared part, we replace the chosen words with [MASK] symbol and feed the new sequence into the shared BERT model again. We use hidden output $h_{[CLS]}$ of token [CLS] as the sentiment representation of the input review, which is referred to as h_{shared}.

3.4.1 Adversarial feature learning

As pointed out in [19], the shared feature space is vulnerable to contamination by domain-specific information. To further ensure the representation h_{shared} of the masked sequences is domain-agnostic, we perform a domain adversarial learning on the shared feature output with a Gradient Reversal Layer (GRL) [6] and a domain classifier.

During the forward propagation, GRL acts as an identity transform, making no changes to those features. During the back-propagation pass, GRL takes the gradient from the subsequent level, reverses the gradient, and passes it to the preceding layer. The gradients from domain classification will not be correctly sent back to the encoder part, thus making it hard to learn domain-distinguishable features. In this way, the reversed gradients from the domain classifier will drive the h_{shared} to contain less domain-specific information and to become more domain-agnostic.

$$h_{grl} = GRL(h_{shared}) \tag{8}$$

where $GRL()$ is the gradient reversal layer, and h_{grl} has the same value as h_{shared} but opposite gradients.

Then, we pass h_{grl} to a domain classifier as follows:

$$\hat{y_d} = softmax(W_{adv_2} tanh(W_{adv_1} h_{grl} + b_{adv_1}) + b_{adv_2}) \tag{9}$$

where $\hat{y_d}$ is the prediction probabilities of domain classification, W_{adv_1} and W_{adv_2} are weights which need to be learned, b_{adv_1} and b_{adv_2} are bias terms.

Given a corpus with N_d training samples for domain classification, the cross-entropy for the prediction is:

$$L_{ds} = \sum_{i=1}^{N_d} \sum_{j=1}^{|D|} y_d^i(j) log(\hat{y_d}^i(j)) \tag{10}$$

where $y_d^i(j)$ is the ground-truth label; $\hat{y_d}^i(j)$ is the prediction probabilities, and $|D|$ is the number of domains.

3.5 Domain-specific sentiment feature extraction

In this part, we use the selected domain-related tokens to learn domain-aware sentence representations for final sentiment classification.

3.5.1 Domain informative feature

Similarly, we can obtain domain-related tokens $\bar{X} = \{x_{j_1}, x_{j_2}, \dots, x_{j_K}\}$ from the private token mask layer, where K is the number of selected domain-related tokens $P_{private}$ in input sequence. Then, hidden representations of those tokens are aggregated as domain-related clue h_j:

$$h_j = \frac{1}{K} \sum_{t=1}^{K} h_{j_t} \tag{11}$$

Besides, we enforce these clues to be domain discriminate with another domain classifier:

$$\hat{y_d} = softmax(W_{dc_2} tanh(W_{dc_1} h_j + b_{dc_1}) + b_{dc_2}) \tag{12}$$

where $\hat{y_d}$ is the prediction probilities of domain classification, W_{dc_1} and W_{dc_2} are weights which need to be learned, b_{dc_1} and b_{dc_2} are bias terms. Similar to Equation 10, we refer to the corresponding cross entropy loss as L_{dp} in this case.

3.5.2 Domain-aware sequence encoding

Since we have the domain-informative clue h_j, we can use it as the query vector and apply the attention mechanism to find the most relevant features of the current review and its corresponding domain. Here, we use simple inner-product attention for simplicity:

$$\alpha_t = softmax(h_j \oplus h_t) \tag{13}$$

$$h_{private} = \sum_{t=1}^{N} \alpha_t * h_t \tag{14}$$

where $\oplus$ means vector concatenation, h_t is the tth token in the input review, and N is the number of tokens in the current review.

3.5.3 Sentiment classification

The final feature for sentiment classification is the concatenation of h_{shared} and $h_{private}$. We use a shared sentiment classifier for all domains and the probability of each sentiment is calculated as follows:

$$h_c = h_{shared} \oplus h_{private} \tag{15}$$

$$\hat{y}_s = softmax(W_{sc_2} tanh(W_{sc_1} h_c + b_{sc_1}) + b_{sc_2}) \tag{16}$$

where $\hat{y}_s$ is the prediction probabilities of sentiment classification, W_{sc_1} and W_{sc_2} are weights which need to be learned, b_{sc_1} and b_{sc_2} are bias terms.

Given N_s training samples for sentiment classification, the cross-entropy for the sentiment prediction is:

$$L_s = \sum_{i=1}^{N_s} \sum_{j=1}^{C} y_s^i(j) log(\hat{y}_s^i(j)) \tag{17}$$

where $y_s^i(j)$ is the ground-truth sentiment label; $\hat{y}_s^i(j)$ is the prediction probabilities, and C is the number of sentiment polarities.

3.6 Final loss

The total loss of our model can be computed as follows:

$$L_{all} = \lambda_{ds} * L_{ds} + \lambda_{dp} * L_{dp} + \gamma * L_s + \beta \|\theta\|^2 \tag{18}$$

where λ_{ds} and λ_{dp} are coefficients for domain classification, γ is coefficients for sentiment classification, and β is the coefficients for L2 regularization.

Table 1 Statistics of datasets from 16 domains

Dataset	Train	Dev.	Test	Avg. length
Books	1400	200	400	159
Electronics	1398	200	400	101
DVD	1400	200	400	173
Kitchen	1400	200	400	89
Apparel	1400	200	400	57
Camera	1397	200	400	130
Health	1400	200	400	81
Music	1400	200	400	136
Toys	1400	200	400	90
Video	1400	200	400	156
Baby	1300	200	400	104
Magazines	1370	200	400	117
Software	1315	200	400	129
Sports	1400	200	400	94
IMDB	1400	200	400	269
MR	1400	200	400	21

4 Experiments

4.1 Dataset

We use the dataset from [19][1] for multi-domain sentiment classification task. which consists of product and movie reviews from 16 domains. Each dataset has roughly 2000 examples. Following previous works, we partition the dataset of each domain into training, development, and testing sets according to the proportions of 70%, 10%, and 20%. The detailed statistics of all the datasets are listed in Table 1. From Table 1, we can see that reviews from different domains have highly variant average lengths. As each domain has a similar number of reviews for training and test, we use accuracy to evaluate the proposed models as in previous works.

4.2 Implementation details

We adopt $\text{BERT}_{\text{base}}$, to be specific, its implementation[2] in PyTorch for all the experiments. The maximum sequence length for the BERT model is set to 128. The mini-batch size is set to 8 and we train the model for 3 epochs. We select hyper-parameters by tuning our model on the development set. The model with the highest averaged accuracy on the development set is chosen for final comparison. Adam is adopted to optimize all our models with an initial learning rate of 0.00001. The coefficients λ_{ds} and λ_{dp} for domain classification loss are set to 0.002 and γ for sentiment classification loss is set to 1. For domain descriptors, the dimension is set to 200. For multi-domain sentiment classification, we train on the domain classification task in all domains for the first 2000 steps and sentiment classification in all domains for the next 3000 steps. After that, we train the model with both sentiment classification and domain classification in all domains jointly. For cross-domain experiments, we train on the domain classification task in all domains for the first 2000 steps and sentiment classification in source domains for the next 3000 steps. After that, we train the model with both sentiment classification and domain classification in source domains jointly. We reported the averaged results of five different seeds for all experiments. We use stop words from this site[3] and sentiment words from [10].

4.3 Multi-domain classification

We experiment with multi-domain sentiment classification on 16 test sets respectively. We compare several baselines and previous state-of-the-art models. All the methods use the same train/valid/test split provided by [19]. And unlabeled reviews from target domains are available for learning domain-invariant sentiment features.

Single Task. We use a bi-directional LSTM and a simple CNN model as single-task baselines which are trained on each domain independently.

BERT [5]. BERT is a pre-trained contextualized representation learning model which has achieved state-of-the-art results on many tasks. We use the pre-trained BERT-base model and fine-tune it for each domain.

ASP-MTL [19]. The model adopted adversarial training on the shared part and separate LSTMs for each domain in the private part.

DA-MTL [33]. It dynamically generated a query vector for each instance and then used this query vector to attend over the hidden representations of the input sentence.

DSR-at [20]. It was also based on the share-private scheme. Different from ASP-MTL, it applied memory networks as the private feature extractor.

MAN-NLL [4]. This model was also based on the share-private scheme and it provided theoretical justifications for the multi-nominal adversarial network.

DAEA [2]. This was an attention-based method that first generates domain-specific query vector and domain-aware word embeddings. It then used the query vector to attend over the hidden representations from BLSTM with domain-aware word embeddings as input.

1 http://pfliu.com/paper/adv-mtl.html.

2 https://github.com/huggingface/transformers.

3 https://github.com/amueller/word_cloud.

DAEA+BERT [2]. It improved DAEA by using BERT as word initialization. It was the previous state-of-the-art model in multi-domain sentiment classification. The domain-wise results were not provided in the original paper and we only report the overall result.

DACL [29]. This method also employed the shared-private structure and deployed dual adversarial regularization to align features across different domains and between labeled and unlabeled data.

GLR-MTL [25]. This work proposed a generic dual channels multi-task learning framework to capture global-shared, local-shared, and private features simultaneously.

MRAN [31]. This method introduced the domain and category mixup regularizations to enrich intrinsic features and consistent predictions.

CAN [30]. This model adopted a conditional domain discriminator to model the domain variance and entropy conditioning to guarantee the transferability of the shared features.

We present results of multi-domain text classification in Table 2. Generally, using data from multiple domains improves average classification performances. We can see that large-scale pretraining helps BERT achieve superior performance on the single domain setting. It even outperforms **ASP-MTL** and **DSR-at** which use labeled data from multiple domains. Our model outperforms all the other models in 10 out of 16 domains and achieves the best performance on average accuracy.

Compared to the previous state-of-the-art **DAEA+BERT** model, our model still achieves 1.91% absolute performance gain on average accuracy. DAEA+BERT model learns a domain-aware sentiment representation of the input review while our model learns better domain-invariant and domain-specific sentiment features. Compared with other shared-private methods (ASP-MTL, MAN-NLL, DACL, CAN, MRAN and GLR-MTL), our model obtains the best results or comparable performances to the best ones in 14 out of 16 domains. We can conclude that the utilization of tokens masking networks helps to pick out domain-specific tokens and acquires better domain-agnostic and domain-aware representations. For domains like Magazines, single BLSTM alone already achieves good performances. Thus, sentiment features from other domains contribute little to final sentiment prediction. While for harder domains like MR, Music, Books, and Electronics, our model brings significant improvements, showing that our model excels at utilizing features shared by different domains than other models.

4.4 Cross-domain experiments

Multi-domain and Cross-domain sentiment classification both aim at transferring sentiment knowledge learned from source domains to target domains. Unlike multi-domain sentiment classification, the task of cross-domain sentiment classification doesn't provide any labeled training data for the target domain. Thus, it calls for better utilization of the shared knowledge across all domains. To further understand whether BERTMasker achieves such capability, we also test our model on the 15-to-1 cross-domain sentiment classification setting [2, 19], where models are trained using the training data of sentiment and domain classification from 15 domains and unlabeled data from the target domain.

As shown in Table 3, our model achieves 3.31% performance gain in averaged accuracy compared to the previous best performing model DAEA. Besides, it outperforms all the other models in 15 out of 16 domains on the cross-domain sentiment classification task. By comparing the performances of these models between Table 2 and Table 3, we can see that DAEA performs extremely well in the video domain on both multi-domain and cross-domain settings. The ASP-MTL model gets 2.3% performance gains in video domain under cross-domain setting than under multi-domain settings while the DSR-at model loses 5% in terms of accuracy. The performance of our model decreases by 2.5% in the video domain and by 1.14% in all domains. We can conclude that our performance drop in the video domain is relatively rational. These results confirm the superiority of token masking networks in BERTMasker, which manifests in learning better shared representations for sentiment classification than other models.

4.5 Ablation test

To further explore how well each component contributes to the prediction of sentiment, we carry out an ablation study of BERTMasker on the test set in the multi-domain setting. As shown in Table 4, the performance decreases when removing either the shared or private network. The removal of the private part leads to more performance loss compared to the shared part. An intuitive explanation is that the domain-aware sentiment features integrate both domain-agnostic and domain-specific sentiment features through the attention mechanism. Moreover, the token masking network in the private part helps increase the performance by 0.32%, proving its effectiveness in choosing domain-informative tokens. By comparing between results of the model *w/o shared part* and *w/o shared mask*, we can see that directly adding shared features without masking even slightly hurts the performance, which proves that masking helps reducing noise in learning transferable features across domains. On the contrary, adding shared features with masking can further improve the performance by 0.43%.

Masking constraints Besides, further experiments on two masking constraints demonstrate both stop words masking and sentiment words masking improve the performance of BERTMasker, by 0.60% and 0.11% respectively. As sentiment words are crucial features for final

Table 2 Results of multi-domain sentiment classification

Domain	Single Domain			Multiple Domains										
	BLSTM*	CNN*	BERT*	ASP-MTL	DA-MTL	DSR-at	MAN-NLL	DACL	GLR-MTL	MRAN	CAN	DAEA*	DAEA+BERT*	BERTMasker
Books	81.00	85.30	87.00	84.00	88.50	89.10	86.80	87.50	88.30	87.00	87.80	89.00	N/A	93.50
Electronics	81.80	87.80	88.30	86.80	89.00	87.90	88.80	90.30	90.30	89.00	91.60	91.80	N/A	94.00
DVD	83.30	76.30	85.60	85.50	88.00	88.10	88.60	89.80	87.30	89.00	89.50	88.30	N/A	88.75
Kitchen	80.80	84.50	91.00	86.20	89.00	85.90	89.90	91.50	89.80	93.00	90.80	90.30	N/A	93.50
Apparel	87.50	86.30	90.00	87.00	88.80	87.80	87.60	89.50	88.20	91.50	87.00	89.00	N/A	91.25
Camera	87.00	89.00	90.00	89.20	91.80	90.00	90.70	91.50	89.50	93.00	93.50	92.00	N/A	92.75
Health	87.00	87.50	88.30	88.20	90.30	92.90	89.40	90.50	90.50	90.00	90.40	89.80	N/A	95.50
Music	81.80	81.50	86.80	82.50	85.00	84.10	85.50	86.30	87.50	86.50	86.90	88.00	N/A	91.75
Toys	81.50	87.00	91.30	88.00	89.50	85.90	90.40	91.30	89.80	86.00	90.00	91.80	N/A	93.50
Video	83.00	82.30	88.00	84.50	89.50	90.30	89.60	88.50	90.80	88.50	88.80	92.30	N/A	91.50
Baby	86.30	82.50	91.50	88.20	90.50	91.70	90.20	92.00	92.30	90.00	92.00	92.30	N/A	93.00
Magazines	92.00	86.80	92.80	92.20	92.00	92.10	92.90	93.80	92.30	93.50	94.50	96.50	N/A	95.25
Software	84.50	87.50	89.30	87.20	90.80	87.00	90.90	90.50	91.80	89.50	90.90	92.80	N/A	95.75
Sports	86.00	85.30	90.80	85.70	89.80	85.80	89.00	89.30	87.80	90.50	91.20	90.80	N/A	93.50
IMDB	82.50	83.30	85.80	85.50	89.80	93.80	87.00	87.30	87.50	89.00	88.50	90.80	N/A	90.50
MR	74.80	79.00	74.00	76.70	75.50	73.30	76.70	76.00	72.70	78.50	77.10	77.00	N/A	84.50
Avg	83.80	84.49	88.16	86.09	88.61	87.86	88.38	89.10	88.52	89.03	89.41	90.16	90.5	92.41

Accuracy (%) is adopted for evaluation. ∗ refers to results taken from [2]. For other models, results are taken from their corresponding papers

Table 3 Results of cross-domain (15-to-1) sentiment classification

	ASP-MTL	DSR-at	DAEA	BERTMasker
Books	81.50	85.80	87.30	90.75
Electronics	83.80	89.50	85.80	94.25
DVD	84.50	86.30	88.80	89.25
Kitchen	87.50	88.30	88.00	91.75
Apparel	85.30	85.80	88.00	91.50
Camera	85.30	88.80	90.00	91.50
Health	86.00	90.50	91.00	94.75
Music	81.30	84.80	86.50	90.25
Toys	88.00	90.30	90.30	93.50
Video	86.80	85.30	91.30	89.00
Baby	86.50	84.80	90.30	93.50
Magazines	87.00	84.00	88.50	90.25
Software	87.00	90.80	89.80	93.00
Sports	87.00	87.00	90.50	94.25
IMDB	84.00	83.30	85.80	91.00
MR	72.00	76.30	75.50	81.75
Avg	84.59	86.35	87.96	91.27

Accuracy (%) is adopted for evaluation

Table 4 Ablation test results of BERTMasker on multi-domain sentiment classification

	Avg. accuracy (%)
w/o shared part	92.0
w/o private part	90.98
w/o shared mask	91.98
w/o private mask	92.09
w/o sentiment word mask	92.3
w/o stop word mask	91.81
Full model	92.41

Average accuracy is presented. *w/o* stands for *without*

sentiment classification, they are less likely to be chosen as domain-specific tokens whether we have the sentiment constraint or not. In contrast, stop words do not directly correlate with final sentiment classification, they maybe be wrongly selected as domain-specific tokens and add noise to the aggregated domain representations. Thus, removing stop words from masking can reduce such noise, purify the masked tokens and improve the interpretability of the remaining domain-invariant texts.

5 Analysis of masking

In this part, we conduct several quantitative and qualitative experiments with BERTMasker on it masking part.

5.1 Number of words masked

As observed in Table 5, the number and percentage of masked tokens of each domain correlate with its average sentence length, where domains with longer average sequence length usually have more tokens masked and lower masking rate. Another interesting finding is that the final masking rates of both shared and private parts are similar to the percentage (15%) of [MASK] token in the Mask Language Model pre-training task of BERT. We leave it as future work to explore whether this rate correlates with the implementation of mask in BERT or the number of domain-related tokens in original data distribution.

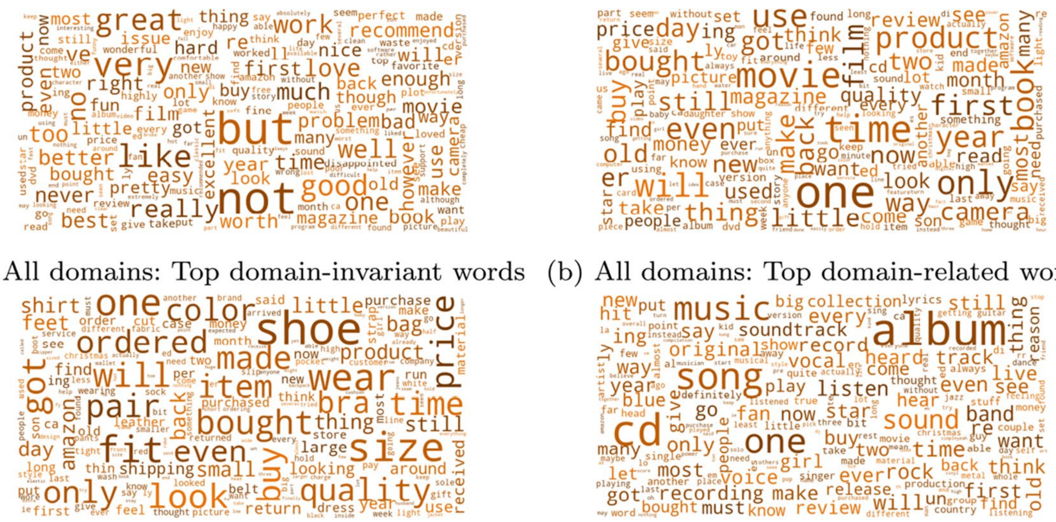
5.2 Top words masked

Apart from the number and percentage of masking, we also would like to investigate whether the token masking networks of BERTMasker mask meaningful words. In Fig. 5a, b, we use word cloud to illustrate tokens after masking from the shared part and masked tokens from the private part of all domains. Besides, we exhibit masked tokens from the private part of Apparel and Music domains. From Fig. 5b we can see that, the top tokens from the masked sequence in the shared part are mostly domain-invariant sentiment-related words, which includes polarity words like *good, great, well*, negation words like *but, not, no* and intensifiers like *really, very*. This demonstrates that after the token masking network removing domain-related tokens, the shared part focuses more on domain-invariant sentiment features. From Fig. 5c, d, we find that as two domains share fewer opinion targets, the distribution of domain-related tokens from their corresponding private masking networks are quite different from each other, where Apparel domain can be depicted with words like *fit, shoes, wear, size, shirt, .etc* and Music domain can be represented using words including *album, song, sound, music, cd, .etc*. When analyzing Fig. 5a, we find that no domain-related words outnumber the other words in the private part from all domains, which again shows the distinction of data distribution of domains in the datasets.

5.3 Domain classification after masking

To further verify whether masking "domain-related" tokens from a text improves its domain invariance, we conduct domain classifications on both original and masked texts. Here, we utilize BERT-base as a powerful feature extractor and apply an MLP similar to that in Eq. 12 for domain classification. We evaluate the results using accuracy.

As shown in Table 6, it's relatively easy to distinguish domains based on original texts. Our mask network successfully degrades the domain classification performance by over 10% on masked texts. This reveals that our strategy



(a) All domains: Top domain-invariant words (b) All domains: Top domain-related words

(c) Apparel: Top domain-related words (d) Music: Top domain-related words

Fig. 5 Word cloud of Tokens from Token Masker Layer. Larger word size means higher frequency of occurrence

Table 5 The number and percentage of masked words in shared and private part of BERTMasker on test set in multi-domain sentiment classification setting

	Shared (no./portion)	Private (no./portion)	Avg. length
Books	39.78/0.21	36.08/0.19	190
Electronics	30.63/0.24	25.93/0.20	128
DVD	38.46/0.17	35.31/0.16	226
Kitchen	26.31/0.24	26.12/0.23	111
Apparel	17.27/0.23	17.17/0.23	74
Camera	33.46/0.23	32.15/0.22	148
Health	23.78/0.23	21.82/0.22	101
Music	33.62/0.21	32.22/0.20	162
Toys	27.49/0.25	25.43/0.23	112
Video	36.51/0.19	30.46/0.16	191
Baby	30.39/0.24	27.02/0.21	128
Magazines	35.16/0.24	32.16/0.22	144
Software	33.10/0.22	29.77/0.20	151
Sports	27.74/0.22	26.63/0.21	125
IMDB	51.78/0.20	46.05/0.17	264
MR	7.48/0.27	6.30/0.23	27
Avg.	30.81/0.22	28.16/0.20	143

Table 6 Results of domain classification on original sequences, sequences after removing masked words and masked words

	Accuracy (%)
On masked sequences	71.0
On original sequences	81.47
On masked words	70.27

of masking is working towards our expectations of domain-invariant text. However, as we don't have direct knowledge of what domain-related tokens are, tokens extracted using the masking network constrained by external sentiment and stop word lexicons are sub-optimal for the domain classification task. Thus, the result demonstrates that the remaining text still contains rich clues for domain classification.

To further explore how the masking works on each domain, we visualize the confusion matrices of domain classification on original and remaining text separately in Fig. 6. For example, by comparing the *Sports* row in Fig. 6a, b, we can see that shallow blocks in 6a become darker in 6b and opposite case happens to darker blocks. This reflects that domain classifier can't find necessary features on the remaining texts, thus mis-classifies more cases into domains sharing some similarities with *Sports* domain, eg. *Electronics*, *Toys*, *Camera* and even on *Software* domain.

From the above experiments, we can see that the token-level masking strategy succeeds in transforming the sentences to be more domain-invariant in the shared part and selecting domain-related words for better domain-aware sentiment feature learning.

6 Case study and error analysis

6.1 Case study

We visualize the words selected by token masking networks in BERTMasker from both shared and private parts in Fig. 7. As illustrated in Example 1, the model successfully masks

(a) On original sequences (b) On masked sequences

Fig. 6 Confusion matrices of domain classification on original and masked sequences

Example 1	
<u>Domain: Baby</u>	True: ☺ Predict: ☺
BERT input	*[CLS] the fabric is soft and cushion , and what ' s most important it provides good support for my baby . she loves it [SEP]*
Shared Input	*[CLS] the **[MASK]** is soft and **[MASK]** , and what ' s most important it provides good support for my **[MASK]** . she loves it [SEP]*
Private Mask	*fabric, cushion, provides, baby*
Example 2	
<u>Domain: Software</u>	True: ☺ Predict: ☺
BERT input	*this world reno ##wn editor is the best when it comes to editing and managing your digital photo [SEP]*
Shared Input	*[CLS] this **[MASK]** **[MASK]** ##wn **[MASK]** is the best when it **[MASK]** to editing and **[MASK]** your **[MASK]** **[MASK]** [SEP]*
Private Mask	*reno ##wn editor editing managing*

Fig. 7 Visualization of masked words in two sentences from magazine and baby domains

domain-related words like *fabric, cushion, baby* in the sentence and makes correct sentiment predictions based on both domain-invariant and domain-aware representations.

However, we note that in many cases, due to the existence of unknown words and errors incurred by the word-piece tokenizer used by BERT, the masked tokens may not be semantically adequate or meaningful. From Example 2, we can see that as *renown* is not recognized by BERT, it further influences the masking result in the shared part. Besides, we notice that in some cases, tokenized negation expressions and sentiment words with different forms (past tense, plurals, etc.) are sometimes wrongly masked. These may lead to failures of the BERTMasker model, especially when there are only limited sentiment words in the short reviews. This suggests that we need better curating of the masking constraints.

6.2 Quantitative error analysis

6.2.1 Manual error analysis

We also perform manual analysis on randomly sampled 20% of mis-classified reviews. We find that in over 63% error cases, the authors wrote about both positive and negative opinions, which shows that reviews with mixed sentiments are generally hard to classify. Among those cases, about 13% of reviews' polarities are derived from the last conclusion sentences. In 37.4% of cases, one sentiment overwhelms the other one. In 3.7% of cases, the authors expressed positive and negative sentiment towards different targets/aspects. In 8.4% of cases, the authors held different sentiments towards the products and content of products, e.g., the books and stories or roles in the books. In 0.93% of cases, the authors described some bad things brought by the good quality of the products, e.g., the camera can take a clear picture of pores in your face. Besides, over 10.2% of cases contain sarcasms or double negations which are not easy to handle. Also, in 6.5% of cases, the authors talked about counterfactual situations relating to the products.

Meanwhile, we conduct automatic analyses to evaluate the influence of the length of reviews, number of sentiment words, and number of negation words.

6.2.2 Influence of review length

The average length of all reviews is 127 (words), while the average length of mis-classified reviews is 147. We see that mis-classified reviews are longer on average. For example, the accuracy on reviews with over 187 words is 89.5% while the accuracy on all reviews is 92.41%. Longer reviews usually contain diverse positive and negative opinions, thus making it difficult for sentiment classifiers to figure out the most salient polarity. During processing, our model truncates the reviews to have less than 200 words, which could potentially harm the performance on longer reviews.

6.2.3 Influence of number of negation words

If a review contains many negation words, it usually means that this review has turning points in sentiments, which makes it hard for models to classify. Our model only gets an accuracy of 87.38% on reviews with over 7 negation words while 92.41% on all reviews, which verifies the above assumption.

6.2.4 Influence of number of sentiment words

Here, we use the *diff* value in Eq. 19 to roughly measure the difference between positive and negative sentiment words.

$$diff = \frac{\left|num_{pos} - num_{neg}\right|}{num_{pos} + num_{neg}} \tag{19}$$

If words of one sentiment outnumber those of the other sentiment in a review, the value would be quite closer to 1 and the polarity of the review would be more likely to be the dominant one. The smaller difference usually means a similar number of positive and negative words in a review, implying that both sentiments are expressed, and complex sentiment semantic composition exists. As a result, our model only gets an accuracy of 92.41% on all reviews while it achieves 89.97% on reviews with a *diff* value less than 0.1.

7 Conclusion

In this paper, we propose the BERTMasker model with token masking networks under the shared-private framework. In the shared part, instead of directly learning domain-variant features in a high-dimensional space, we propose to first transform sentences to be more domain-invariant through masking domain-related words. Then BERTMasker learns a good sentiment representation from the remaining domain-invariant review, which utilizing a similar input format of BERT's mask language model pretraining. In the private part, BERTMasker aggregates the masked domain-related tokens as the domain representation and acquires a domain-aware sentiment representation. Our model outperforms existing works on the benchmark dataset by a large margin in both multi-domain and cross-domain settings. Detailed analysis of the masked words further proves the effectiveness of our proposed masking strategy.

In the future, we would like to work in following directions: (1) replace the mask network with a simpler network, e.g. distilled BERT models, to accelerate training and inference of our model. (2) incorporate more external knowledge to guide the fine-grained and accurate selection of domain-related words and phrases. (3) explore whether replacing certain portion of [MASK] with other random words could improve robustness of the proposed method.

References

1. Bousmalis K, Trigeorgis G, Silberman N, Krishnan D, Erhan D (2016) Domain separation networks. In: Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R (eds) Advances in neural information processing systems, vol 29. Curran Associates, Inc., pp 343–351. http://papers.nips.cc/paper/6254-domain-separation-networks.pdf. Accessed 30 Mar 2021
2. Cai Y, Wan X (2019) Multi-domain sentiment classification based on domain-aware embedding and attention. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI-19. International joint conferences on artificial intelligence organization, , pp 4904–4910. https://doi.org/10.24963/ijcai.2019/681
3. Chen X, Awadallah AH, Hassan H, Wang W, Cardie C (2019) Multi-source cross-lingual model transfer: learning what to share. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy, pp 3098–3112. https://www.aclweb.org/anthology/P19-1299. Accessed 30 Mar 2021
4. Chen X, Cardie C (2018) Multinomial adversarial networks for multi-domain text classification. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers). Association for Computational Linguistics, New Orleans, Louisiana, pp. 1226–1240. https://doi.org/10.18653/v1/N18-1111. https://www.aclweb.org/anthology/N18-1111. Accessed 30 Mar 2021
5. Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, volume 1 (long and short papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423. https://www.aclweb.org/anthology/N19-1423. Accessed 30 Mar 2021
6. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):2030–2096
7. Guo J, Shah D, Barzilay R (2018) Multi-source domain adaptation with mixture of experts. In: Proceedings of the 2018 conference on empirical methods in natural language processing. Association for Computational Linguistics, Brussels, Belgium, pp 4694–4703.

https://doi.org/10.18653/v1/D18-1498. https://www.aclweb.org/anthology/D18-1498. Accessed 30 Mar 2021
8. Howard J, Ruder S (2018) Universal language model fine-tuning for text classification. In: Proceedings of the 56th annual meeting of the Association for Computational Linguistics (volume 1: long papers). Association for Computational Linguistics, Melbourne, Australia, pp 328–339. https://doi.org/10.18653/v1/P18-1031. https://www.aclweb.org/anthology/P18-1031. Accessed 30 Mar 2021
9. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, KDD '04. Association for Computing Machinery, New York, NY, USA, pp 168-177. https://doi.org/10.1145/1014052.1014073
10. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp 168–177
11. Ilyas A, Santurkar S, Tsipras D, Engstrom L, Tran B, Madry A (2019) Adversarial examples are not bugs, they are features. In: Advances in neural information processing systems, vol 32. Curran Associates, Inc, pp 125–136
12. İrsoy O, Cardie C (2014) Opinion mining with deep recurrent neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)Association for Computational Linguistics, Doha, Qatar, pp 720–728. . https://doi.org/10.3115/v1/D14-1080. https://aclanthology.org/D14-1080
13. Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE (1991) Adaptive mixtures of local experts. Neural Comput 3(1):79–87. https://doi.org/10.1162/neco.1991.3.1.79
14. Jang E, Gu S, Poole B (2016) Categorical reparameterization with gumbel-softmax. arXiv:1611.01144
15. Ke P, Ji H, Liu S, Zhu X, Huang M (2019) Sentilr: linguistic knowledge enhanced language representation for sentiment analysis. arXiv:1911.02493
16. Li S, Zong C (2008) Multi-domain sentiment classification. In: Proceedings of ACL-08: HLT, short papers. Association for Computational Linguistics, Columbus, Ohio, pp 257–260. https://www.aclweb.org/anthology/P08-2065. Accessed 30 Mar 2021
17. Liu B (2012) Sentiment analysis and opinion mining. Synth Lect Hum Lang Technol 5(1):1–167
18. Liu P, Joty S, Meng H (2015) Fine-grained opinion mining with recurrent neural networks and word embeddings. In: Proceedings of the 2015 conference on empirical methods in natural language processing. Association for Computational Linguistics, Lisbon, Portugal, pp 1433–1443. https://doi.org/10.18653/v1/D15-1168. https://aclanthology.org/D15-1168. Accessed 30 Mar 2021
19. Liu P, Qiu X, Huang X (2017) Adversarial multi-task learning for text classification. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers). Association for Computational Linguistics, Vancouver, Canada, pp 1–10. https://doi.org/10.18653/v1/P17-1001. https://www.aclweb.org/anthology/P17-1001. Accessed 30 Mar 2021
20. Liu Q, Zhang Y, Liu J (2018) Learning domain representation for multi-domain sentiment classification. In: Proceedings of the 2018 conference of the North American Chapter of the Association for computational linguistics: human language technologies, volume 1 (long papers). Association for Computational Linguistics, New Orleans, Louisiana, pp 541–550. https://doi.org/10.18653/v1/N18-1050. https://www.aclweb.org/anthology/N18-1050
21. McCann B, Bradbury J, Xiong C, Socher R (2017) Learned in translation: contextualized word vectors. In: Advances in neural information processing systems, pp 6294–6305
22. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the 2002 conference on empirical methods in natural language processing (emnlp 2002). Association for Computational Linguistics, pp 79–86. https://doi.org/10.3115/1118693.1118704. https://www.aclweb.org/anthology/W02-1011. Accessed 30 Mar 2021
23. Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers). Association for Computational Linguistics, New Orleans, Louisiana, pp 2227–2237. https://doi.org/10.18653/v1/N18-1202. https://www.aclweb.org/anthology/N18-1202. Accessed 30 Mar 2021
24. Qiu G, Liu B, Bu J, Chen C (2009) Expanding domain sentiment lexicon through double propagation. In: Proceedings of the 21st international joint conference on artificial intelligence, IJCAI'09. Morgan Kaufmann Publishers Inc., San Francisco, pp 1199–1204
25. Su X, Li R, Li X (2020) Multi-domain transfer learning for text classification. In: Zhu X, Zhang M, Hong Y, He R (eds) Natural language processing and Chinese computing. Springer International Publishing, Cham, pp 457–469
26. Sun C, Huang L, Qiu X (2019) Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In: Proceedings of the 2019 conference of the North American Chapter of the Association for Computational Linguistics: human language technologies, volume 1 (long and short papers). Association for Computational Linguistics, Minneapolis, Minnesota, pp 380–385. https://doi.org/10.18653/v1/N19-1035. https://www.aclweb.org/anthology/N19-1035. Accessed 30 Mar 2021
27. Tian H, Gao C, Xiao X, Liu H, He B, Wu H, Wang H, Wu F (2020) SKEP: sentiment knowledge enhanced pre-training for sentiment analysis. In: Proceedings of the 58th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Online, pp 4067–4076. https://doi.org/10.18653/v1/2020.acl-main.374. https://www.aclweb.org/anthology/2020.acl-main.374. Accessed 30 Mar 2021
28. Wu F, Huang Y (2015) Collaborative multi-domain sentiment classification. In: 2015 IEEE international conference on data mining. IEEE, pp 459–468
29. Wu Y, Guo Y (2020) Dual adversarial co-learning for multi-domain text classification. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, no 04, pp 6438–6445. https://doi.org/10.1609/aaai.v34i04.6115. https://ojs.aaai.org/index.php/AAAI/article/view/6115. Accessed 30 Mar 2021
30. Wu Y, Inkpen D, El-Roby A (2021) Conditional adversarial networks for multi-domain text classification. In: Proceedings of the second workshop on domain adaptation for NLP. Association for Computational Linguistics, Kyiv, Ukraine, pp 16–27. https://aclanthology.org/2021.adaptnlp-1.3. Accessed 30 Mar 2021
31. Wu Y, Inkpen D, El-Roby A (2021) Mixup regularized adversarial networks for multi-domain text classification. In: IEEE international conference on acoustics, speech and signal processing, ICASSP 2021, Toronto, ON, Canada, June 6–11, 2021. IEEE, pp 7733–7737. https://doi.org/10.1109/ICASSP39728.2021.9413441
32. Yuan J, Wu Y, Lu X, Zhao Y, Qin B, Liu T (2020) Recent advances in deep learning based sentiment analysis. Sci China Technol Sci 63(10):1947–1970. https://doi.org/10.1007/s11431-020-1634-3
33. Zheng R, Chen J, Qiu X (2018) Same representation, different attentions: shareable sentence representation learning from multiple tasks. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI-18. International joint conferences on artificial intelligence organization, pp 4616–4622. https://doi.org/10.24963/ijcai.2018/642