A Multi-Task Learning Approach to Improve Sentiment Analysis with Explicit Recommendation

Olivier Habimana, Yuhua Li*, Ruixuan Li*, Xiwu Gu, Yuqi Peng
School of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan, China
{habolivier, idcliyuhua, rxli, guxiwu, pangyoki}@hust.edu.cn

***Abstract*—When expressing sentiment towards products, customers often explicitly indicate their recommendation status. Nevertheless, most existing literature focuses on sentiment analysis but neglects the rich correlation information that may be brought by explicit recommendation classification. We argue that the two tasks are correlated and hence, the knowledge in explicit recommendation classification can also be beneficial to sentiment analysis. Consequently, in this paper, a novel bidirectional encoder representations from transformers (BERT)-enhanced multi-task learning (BeMTL) approach is proposed to improve sentiment analysis with explicit recommendation classification. Specifically, the proposed MTL approach takes contextualized word embeddings produced by the pre-trained BERT-based embedding layer. Then, it learns the sentence contextual features shared between both tasks with a convolutional multi-head attention neural network. To fully exploit the correlation information between sentiment analysis and explicit recommendation classification tasks, a novel inter-task matching layer (IML) is designed to match their representations. In nutshell, our study reveals the potential of multi-task learning models on such types of problems, and experimental results on two Amazon datasets show that our approach outperforms the state-of-the-art baseline approaches for sentiment analysis.**

***Index Terms*—sentiment analysis, explicit recommendation classification, multi-task learning, deep learning, convolutional neural network, multi-head attention**

I. Introduction

Recently, with the advent of e-commerce platforms, customers often freely express their ideas and thoughts about products, as a result of which there is an abundance of user-generated reviews (UGRs). Analyzing and getting the hidden insight into UGRs has become a de-facto skillset for many organizations, notably, in marketing strategies as the *online word of the mouth* has a strong weight in customers [1]. In analysis of UGRs, the primary task is sentiment analysis [2], which aims to determine whether a text bears a positive polarity or a negative one. Besides, there are other auxiliary tasks including review explicit recommendation classification [3], [4], which aims to classify whether a review text explicitly recommends the reviewed product or not.

When expressing their opinions towards products, customers often explicitly recommend them or not to indicate their level of satisfaction [4]. Mostly, explicit recommendations are expressed towards certain aspects of the product. Consider the following review texts taken from the Amazon Women Clothes dataset. (a) "*I love this top. I wear it all the time b'se I like it. The problem is that you can tell I wear it all the time as the fabric has started to fade. I'd still recommend it as it is so comfortable.*" This review text carries positive sentiment polarity and is explicitly recommending the reviewed top. (b) "*I got this dress in hopes of having a really nice winter formal dress. It was not well made at all! The lining didn't line up with the top layer, and the waist puffed out in uneven places. The fabric itself is very nice, but just not well made. I do not recommend this dress.*" Review text (b) contains a negative sentiment polarity and does not recommend the reviewed bra. The above review texts can well illustrate the correlation between customers' opinions and their explicit recommendations. Customers' explicit recommendations expressed towards the products, which can be defined as declarations that these products are suitable for others, proved to be more persuasive than sentiment towards them [1]. This occurs because explicit endorsers are perceived to have a high degree of expertise.

Currently, most existing literature focuses on sentiment analysis task [5] but ignores the rich correlation information that may exist in explicit recommendation classification task [3], [4]. However, from review texts (a) and (b), one can observe that the two tasks are correlated, i.e., users often explicitly use recommending words to emphasize their positiveness and non-recommending words to stress their negativeness. Therefore, if a review text can be categorized as recommending, then its sentiment can be assumed positive. On the other hand, if it can be classified as non-recommending, then it can be considered negative. Consequently, this would improve performance in sentiment analysis. Therefore, from a research viewpoint, it becomes a challenge of whether and how one may benefit when addressing such correlated tasks and how one can share the knowledge from one task to another during the training process.

To this end, we propose to formulate sentiment analysis and explicit recommendation classification problems as a multi-task learning problem and hence build simple yet effective multi-task learning (MTL) approach, which simultaneously optimizes both tasks to improve the performance. In recent years, the MTL [6] scheme has gained popularity mainly because it allows each task to serve as an effective regularization method for the other. In this way, it can potentially

*Corresponding Authors:{idcliyuhua, rxli}@hust.edu.cn

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

make models less prone to overfitting, and hence to improve the performance [7]. The advantages of the MTL have been validated in other closely related tasks [8]–[11], and our tasks are similarly related. Furthermore, the MTL has been recently explored in more loosely related tasks like machine translation and syntactic parsing [12].

In contrast to most existing MTL approaches that divide the layers of a model into shared and task-specific layers, a novel inter-task matching layer (IML) is introduced to better exploit the interaction between sentiment and explicit recommendation classification tasks. To further improve the results, the proposed MTL leverages transfer learning through pre-trained bidirectional encoder representations from transformers (BERT) [13], which is the recent successful contextualized language model.

In summary, our main contributions are as follows:

1) We propose a BERT-enhanced Multi-task Learning approach (BeMTL) to explore how jointly learning sentiment and explicit recommendation classification tasks can improve the performance in sentiment analysis.
2) A convolutional multi-head attention network is brought up to learn sentence contextual features shared between the two tasks.
3) To better exploit the rich correlation information between both tasks, we design an inter-task matching layer (IML) to match their specific-task representations.
4) We extract the knowledge from the pre-trained BERT model to leverage the features extracted by our model.
5) The comprehensive experiments on two real-world datasets that contain review texts with sentiment and explicit recommendation tags show that the proposed BeMTL model strongly outperforms the state-of-the-art approaches across both tasks.

The remainder of the work is organized as follows. First, a summary of related work is given in Section II. Second, the proposed method is described in Section III. Third, we present the experimental setup and results obtained by our model in Section IV. Finally, the conclusion is drawn in Section V.

II. Related Work

A. Single-Task Learning (STL) Methods

A plethora of STL methods have been put forward in sentiment analysis, which is considered as the main task in user review analysis. Short CNN-based models have been proposed to extract sentence local contextual features [14], [15]. However, they proved to be limited in handling word interaction features beyond the window size. To tackle the issue, very deep CNN models that help to deal with global word interaction features have been suggested [16], [17].

Similarly, to deal with global word interaction features, Agarap et al. [3] devised a bidirectional long-short term memory (BiLSTM) that handles sentiment analysis and explicit recommendation classification as separate tasks. Mousa et al. [18] suggested a contextual BiLSTM that models the right and the left contextual features of a word in a sentence. Motivated by the advantage of the attention mechanism to prioritize important features [19], Yang et al. [20] proposed a GRU attention-based network that hierarchically learns the words and sentences of a document. Likewise, Wu et al. [21] introduced an LSTM attention-based model that enhances review representation with the user and product information. However, these recurrent-based models proved to be difficult to parallelize and not able to model global word interaction features at a possible extent. Following the success of self-attention [22], a directional self-attention network [23], which applies one or multiple positional masks, was recently introduced. Despite the success of STL methods in sentiment analysis, they ignore the richness of information that may exist in its related tasks, explicit recommendation classification in particular, which has been overlooked, because only work [3] is found in literature.

B. Multi-Task Learning Methods

Multi-Task Learning (MTL) [6] is a learning approach in machine learning, which stems from the idea that leveraging rich correlation information available in several related tasks helps to improve their generalization performance. Recently, efforts have been made to apply MTL in sentiment analysis and its associated tasks. Zhang et al. [9] designed an MTL framework that uses BiLSTM to deal with shared features and exploit label propagation between related tasks. Majumder et al. [8] suggested an MTL model that jointly learns the sentiment classification and sarcasm detection by using a GRU with attention as a feature extractor. To realize fine-grained sentiment analysis, Balikas et al. [10] designed an MTL that uses LSTM to learn the shared features between ternary and five-class classification. Similarly, Dai et al. [11] adopted an MTL framework to simultaneously learn the categories of a document by using multi-head attention to extract document-shared and category-specific features.

Recently, there has been a growing interest in incorporating sentiment analysis into recommender systems (RS) to enhance explanation of the generated recommendations [24], [25]. The proposed solutions jointly learn in an MTL setting the opinionated contents and user preferences that are implicit recommendations generated by RS. However, to dispel the doubts, it is worth mentioning that our work is different from the mentioned as we aim to improve sentiment analysis of user review texts with explicit recommendation classification via an MTL setting. And, they cannot be compared to our work since they address implicit recommendation whereas we address explicit recommendation. After all, to the best of our knowledge, no deep learning model has been suggested to learn sentiment analysis and explicit recommendation classification tasks simultaneously.

III. Proposed Method

In this section, we introduce the multi-task learning of sentiment analysis and explicit recommendation classification.

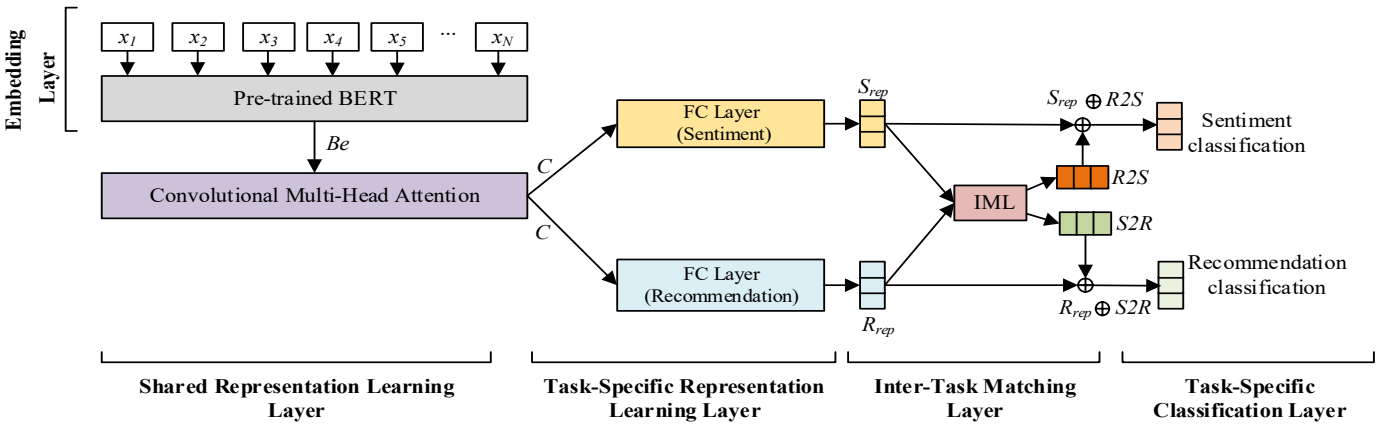


Fig. 1. Overall Architecture of BERT-enhanced Multi-task Learning Approach (BeMTL).

A. Task Definition

We argue that in order to obtain good sentiment analysis results, there is a need to consider the knowledge in explicit recommendation classification task. Thus, in this work we give the following analogy to incorporate this knowledge in explicit recommendation classification task into sentiment analysis. Given an input sentence S with length N, $S = [x_1, x_2, x_3, x_4, ..., x_N]$ where x_i corresponds to the i^{th} word in the sentence S. The sentence S has a sentiment tag (positive/negative) and an explicit recommendation tag (yes/no). We claim that jointly training both tasks and exploiting their correlation information can help to improve the results.

B. Model Overview

To deal with the above-described problem, we propose a BERT-enhanced MTL (BeMTL), which is shown in Fig. 1. The BeMTL consists of five major components that jointly work in the following fashion. BeMTL receives the input sentence $S = [x_1, x_2, x_3, x_4, x_5, ..., x_N]$ and then the embedding layer produces the contextualized word embeddings Be. Afterwards, the shared representation learning layer made by a convolutional multi-head attention network (CNN-MHA) learns shared sentence contextual features, i.e., local and global contextual features, and produces the high-level representation C. To accommodate both tasks, the C representation is fed to the task-specific representation learning layers. An inter-task matching layer (IML) made by coattention is used to match the representations of both tasks, i.e., S_{rep} and R_{rep}. Lastly, the model applies two separate task-specific softmax classifiers that compute the predictions of the sentence for both tasks.

C. Word Embedding Layer

Word embedding layer receives the input sentence $S = [x_1, x_2, x_3, x_4, x_5, ..., x_N]$ where x_i corresponds to the i^{th} word in the sentence S of length N and then maps each word in the sentence S to a high-dimensional vector space through the pre-trained BERT language model [13]. We use pre-trained BERT as a feature-based approach rather than a fine-tuning approach since the former helps to mitigate the out-of-memory issues. Consequently, the output of the embedding layer is a matrix $Be = [Be_1, Be_2, Be_3, Be_4, Be_5, ..., Be_N] \in \mathbb{R}^{d \times N}$, where $Be_i \in \mathbb{R}^d$ corresponds to the contextual vector of the word i in sentence S and d is the embedding dimension.

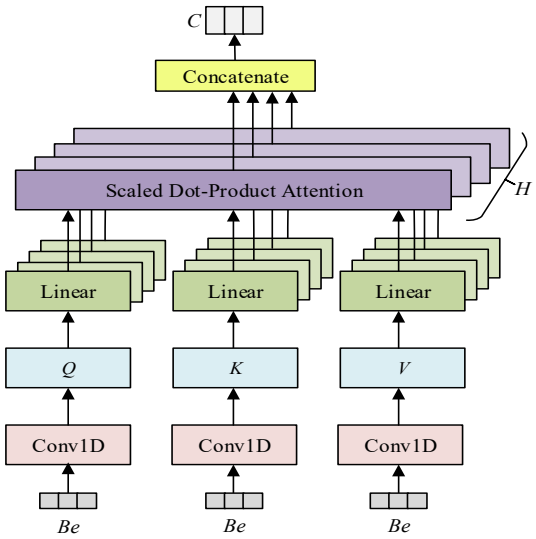


Fig. 2. Convolutional Multi-head Attention (CNN-MHA).

D. Shared Representation Learning Layer

After encoding words of a sentence into contextualized representations, we apply a convolutional multi-head attention network (CNN-MHA) to encode the sentence contextual features shared between our tasks. Therefore, this section first introduces the standard multi-head attention, and then proceeds to the details of our CNN-MHA shown by Fig. 2.

1) Multi-Head Attention: Self-Attention networks (SANs) [22], [23] have received a substantial amount of attention because they are more parallelizable in computation. Moreover, SANs have presented the capability of modeling long-range dependencies by attending all positions of an input sentence regardless of the distance between them. Consequently, to deal with global word interaction features, we adopt

multi-head attention [22], which applies different attention heads to model information at different input positions. Given $Be = [Be_1, Be_2, Be_3, Be_4, Be_5, ..., Be_N] \in \mathbb{R}^{d \times N}$, the input embedding representation, it is first divided into query embeddings Q, key embeddings K, and value embeddings V. Formally, $[Q, K, V] \in \mathbb{R}^{d \times N}$ are expressed as follows:

$$Q, K, V = BeW_Q, BeW_K, BeW_V \tag{1}$$

where W_Q, W_K and $W_V \in \mathbb{R}^{d \times d}$ are parameter matrices and d is the embedding dimension. To apply multi-head attention, the representations [Q,K,V] are transformed into H subrepresentations, such that $Q^h, K^h, V^h \in \mathbb{R}^{\frac{d}{H} \times N}$. Afterwards, each triplet of subrepresentations is supplied to its own scaled dot product attention function where it produces the output vector $C^h = [c_1^h, c_2^h, c_3^h ..., c_N^h]$, with each element c_i^h given by:

$$c_i^h = ATT(q_i^h, K^h)V^h \in \mathbb{R}^{\frac{d}{H}} \tag{2}$$

where ATT(.) is the dot-product attention, which proved to be fast and more memory-efficient [22]. Thus, the results from different attention heads are concatenated to make the final output C expressed as follows:

$$C = [C^1, C^2, C^3, ..., C^H] \in \mathbb{R}^{d \times N} \tag{3}$$

Although multi-head attention has shown the capability to deal with global contextual features, it still presents limited competence when it comes to model local contextual features since it treats the input sentence as a bag-of-words.

2) Convolutional Multi-Head Attention (CNN-MHA): Convolutional combined with self-attention has been recently applied in question answering [26] and neural machine translation [27]. Thus, motivated by the success of this combination, we also adopt it to extract the sentence local and global contextual features shared between our tasks.

We use CNN to learn local contextual features. Our CNN applies a filter with weight matrix $F \in \mathbb{R}^{d \times n}$ on a window of n words to the input embeddings $Be = [Be_1, Be_2, Be_3, Be_4, Be_5, ..., Be_N] \in \mathbb{R}^{d \times N}$. In this way, each word representation in the considered window is enriched with local contextual features of neighboring words. Unlike the standalone multi-head attention described in Subsection III-D1 that directly receives the embedding representation Be, our multi-head attention is instead conditioned to be bounded in the window of n words. Thus, in this setting, the values of Q, K and V in equation (1) are newly expressed as follows:

$$\begin{aligned} Q &= ReLu(Conv1D(Be, F_q) + b_q) \\ K &= ReLu(Conv1D(Be, F_k) + b_k) \\ V &= ReLu(Conv1D(Be, F_v) + b_v) \end{aligned} \tag{4}$$

where $Conv1D(Be, F)$ is a $1D$ convolution operation, which receives Be as input and then applies the filter F. The representations $\{Q, K, V, Be\} \in \mathbb{R}^{d \times N}, \{F_q, F_k, F_v\} \in \mathbb{R}^{d \times n}$, and bias vectors $\{b_q, b_k, b_v\} \in \mathbb{R}^d$.

Finally, with the new Q, K and V expressed in equation (4), we apply the same procedure described in multi-head attention (Subsection III-D1) to compute equation (2) and our final output $C = [C^1, C^2, C^3, ..., C^H] \in \mathbb{R}^{d \times N}$, which contains both local and global contextual features from different heads applied in different convolutional windows.

E. Task-Specific Representation Learning Layer

Our model shares its parameters across sentiment analysis and explicit recommendation classification tasks. Thus, to accommodate the two tasks, the shared features $C = [C^1, C^2, C^3, ..., C^H]$ produced by the CNN-MHA network are fed to two different branches of fully connected layers (FC), which learn task-specific features. Thus, sentiment and explicit recommendation classification task-specific layers produce the task-specific representations $S_{rep} \in \mathbb{R}^{d' \times N}$ and $R_{rep} \in \mathbb{R}^{d' \times N}$, respectively.

F. Inter-Task Matching Layer

Exploiting task relationships have been proved as a way to improve the performance of MTL models [8]. Thus, to match sentiment analysis specific representation S_{rep} and explicit recommendation classification specific representation R_{rep}, we design a novel inter-task matching layer (IML), which utilizes the coattention commonly applied in question answering [26], [28]. We first use the dot product to compute the similarity matrix M, which shows how well S_{rep} and R_{rep} representations, semantically match as is in the following formula.

$$M = S_{rep} \cdot R_{rep}^T \in \mathbb{R}^{N \times N} \tag{5}$$

Next, the row-wise operations are applied to produce recommendation_to_sentiment matching matrix ($R2S$) as follows.

$$\begin{aligned} H_r &= tanh(W_r M^T) \\ \alpha_r &= softmax(w_r^T \cdot H_r) \\ R2S &= S_{rep} \cdot \alpha_r \end{aligned} \tag{6}$$

where $W_r \in \mathbb{R}^{d'' \times N}$, $w_r \in \mathbb{R}^{d''}$. $\alpha_r \in \mathbb{R}^N$ is the attention vector that computes the importance degree of all words in S_{rep}. $R2S \in \mathbb{R}^{d' \times N}$ contains information from the recommendation that is relevant to the sentiment classification task.

Simultaneously, we apply the column-wise operations to compute sentiment_to_recommendation matching matrix ($S2R$) as is expressed in the formula below.

$$\begin{aligned} H_s &= tanh(W_s M) \\ \alpha_s &= softmax(w_s^T \cdot H_s) \\ S2R &= R_{rep} \cdot \alpha_s \end{aligned} \tag{7}$$

where $W_s \in \mathbb{R}^{d'' \times N}$, $w_s \in \mathbb{R}^{d''}$. $\alpha_s \in \mathbb{R}^N$ is the attention vector that computes the importance degree of all words in R_{rep}. $S2R \in \mathbb{R}^{d' \times N}$ contains information from the sentiment that is relevant to explicit recommendation classification.

G. Task-Specific Classification Layer

We use two different softmax layers for classification.

TABLE I
SUMMARY STATISTICS OF THE DATASETS.

Dataset	#Train	#Val	#Test	#Classes	L.Cons
A_WClothes	14091	4697	4698	2	81.09%
A_Electronics	19557	6519	6520	2	91.29%

1) Sentiment Classification: We apply a sentiment-specific softmax layer with size C (C=2) on the sentence representation $S_{rep} \oplus R2S$ as follows:

$$\begin{aligned} P_{sen} &= softmax((S_{rep} \oplus R2S)W^s_{sen} + b^s_{sen}) \\ \hat{y}_{sen} &= \underset{j}{argmax}(P_{sen}[j]) \end{aligned} \tag{8}$$

where $\oplus$ means concatenation, and W^s_{sen} and b^s_{sen} stand for bias and weight for class c, respectively. P_{sen} is the probability distribution, $\hat{y}_{sen}$ is the estimated sentiment class label and j is the actual sentiment class label (0 for negative and 1 for positive).

2) Explicit Recommendation Classification: We use an explicit recommendation-specific softmax layer with size C (C=2) on sentence representation $R_{rep} \oplus S2R$ as follows:

$$\begin{aligned} P_{rec} &= softmax((R_{rep} \oplus S2R)W^r_{rec} + b^r_{rec}) \\ \hat{y}_{rec} &= \underset{j}{argmax}(P_{rec}[j]) \end{aligned} \tag{9}$$

where $\oplus$ means concatenation, and W^r_{rec} and b^r_{rec} stand for bias and weight for class c, respectively. P_{rec} is the probability distribution, $\hat{y}_{rec}$ is the estimated recommendation class value while j is the actual explicit recommendation class label (0 for no and 1 for yes).

H. Multi-Task Training Procedure

To simultaneously train the proposed MTL model on sentiment analysis and explicit recommendation classification tasks, we apply binary cross-entropy objective functions (L_*; * is sen or rec) defined by the equation (10). We tie the two objective functions, i.e., L_{sen} and L_{rec} in an elegant manner where they are given equal priority.

$$L_* = -\sum_{i=1}^{C} t_c(y_{*i}) log \hat{y}_{*i} \tag{10}$$

where C is the number of classes (C=2, in our case of binary classification). $t_c(y_{*i})$ is a one-hot vector representing the distribution of the actual sentiment label or explicit recommendation label and $\hat{y}_{*i}$ is the estimated class value for either sentiment label or explicit recommendation label.

IV. EXPERIMENTS

A. Dataset Description and Evaluation Metric

We test the effectiveness of the BeMTL on two Amazon product review datasets that are available on the Kaggle website. The two datasets present the advantage of having sentiment tags (0 for negative and 1 for positive) and explicit recommendation tags (0 for no and 1 for yes), which makes them suitable for multi-tasking. Table I gives a brief description of their statistics. L.Cons stands for consistency of the labels across the two tasks. The first dataset is Women's E-Commerce Clothing Reviews[1], which contains review texts related to women's clothes. The second dataset is Consumer Reviews of Amazon ProductsA_WClothes[2], which contains review texts about different types of equipment. We label these datasets as A_WClothes and A_Electronics.

We evaluate the skills of the proposed model using the Area under the ROC Curve (AUC) evaluation metric, which takes into account the labels' imbalance. In short, AUC measures the probability that a random positive sample will have a higher score than a random negative sample.

B. Experimental Settings

The inputs to the proposed model are contextualized embeddings produced by pre-trained BERT-BASE[3] (Subsection III-C). During the training, we did not fine-tune the produced embeddings. For both datasets, we set the sentence length (N) to 350. For convolutional operation (Subsection III-D2), we use one dimensional CNN with 100 filters and the kernel window size equal to 3. We use the rectifier linear unit (ReLU) activation function to the convolutional layer. For the multi-head attention layer (Subsection III-D1), the number of attention heads H is fixed to 4. While we train the proposed model on both datasets, the number of epochs varies between $(7, 25)$. For each iteration of the training process, we fix the batch size to 128. While training the proposed model, we minimize the binary cross-entropy losses L_{sen} and L_{rec} expressed by equation (10). Adam optimizer [29] with default parameters is used to update the parameters of both loss functions.

C. Baseline Methods

To prove the superiority of our model, we compare it with the state-of-the-art single task learning (STL) and multi-task learning (MTL) deep learning methods for sentiment analysis.

1) Single-task learning (STL) Methods:

- CNN-Multi [14]: A model that uses two CNN channels with different filters to learn local contextual features.
- VDCNN [17]: A state-of-art very deep CNN proposed in sentiment analysis to deal with long-range dependencies.
- BiLSTM [3]: A bidirectional LSTM model, which is a state-of-art published on the A_WClothes dataset.
- HAN [20]: A hierarchical network with attention model, which has shown strong performance of various review text datasets.
- DiSAN [23]: A recent directional self-attention network, which has achieved promising results in various NLP tasks including sentiment analysis.

2) Multi-Task Learning (MTL) Methods:

- MTLFS [10]: A multi-task learning based on LSTM designed to realize fine-grained sentiment analysis

[1]https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews
[2]https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews
[3]https://github.com/google-research/bert

TABLE II
EXPERIMENTAL RESULTS [IN AUC] OF BEMTL AGAINST STATE-OF-THE-ART MODELS.

		A_WClothes		A_Electronics	
Type	**Model**	**Sentiment**	**Recom**	**Sentiment**	**Recom**
STL	CNN-Multi	76.08%	78.27%	79.81%	70.33%
	VDCNN	75.50%	78.26%	79.18%	74.44%
	BiLSTM	79.16%	83.39%	81.85%	74.12%
	HAN	80.05%	82.37%	82.19%	74.63%
	DiSAN	82.15%	83.53%	82.53%	76.31%
MTL	MTLFS	80.01%	82.26%	81.37%	70.88%
	MMAM	82.22%	83.57%	83.43%	76.21%
	MTLSS	82.57%	85.08%	84.01%	77.33%
Ours	**BeMTL**	**85.59%**	**88.19%**	**86.78%**	**83.22%**

- MMAM [11]: A multi-task multi-attention memory model, which applies the LSTM with a multi-head attention model for fine-grained sentiment analysis
- MTLSS [8]: An MTL model designed for sentiment analysis and sarcasm classification. It uses GRU with attention to learn the shared features and applies fusion method to exploit the relationship between both tasks.

It is worth mentioning, however, that, in our study, we do not compare our model with BERT [13], as it is a large language model with more than 110 million parameters while our model has a maximum of 3.4 million parameters.

D. Model Comparison with Baseline Methods

The experimental results achieved by our BeMTL with a comparison against the baseline methods are presented in Table II (Recom means explicit recommendation). In general, the proposed BeMTL outperforms the baseline models by a large margin across both datasets. On the A_WClothes dataset, results indicate that BeMTL achieves 3.02% and 3.11% AUC scores absolute improvement for sentiment and explicit recommendation classification tasks, respectively. Similarly, on the A_Electronics dataset, BeMTL improves the performance by 2.77% and 5.89% AUC scores respectively in these two respects.

1) Comparison with Single-Task Learning (STL) Methods: In comparison against CNN-based baseline models, i.e., CNN-Multi and VDCNN, we observe that BeMTL outperforms them by a noticeable margin on all the datasets. Therefore, these results evidence that BeMTL is more efficient in modeling sentence contextual features compared with CNN models, which only rely on local contextual features. Particularly, the performance increase reveals the advantages of extracting global word interaction features with multi-head attention.

Besides, compared with RNN and traditional attention-based models, i.e., BiLSTM and HAN, the results imply that the BeMTL model significantly improves the performance of all datasets across the two tasks. Thus, the better performance of the BeMTL model is attributed to the complimentary of both types of contextual features extracted by our proposed CNN-MHA model. In comparison with DiSAN, which is based on self-attention like our model, BeMTL sill outperforms it on all datasets.

TABLE III
ABLATION STUDY RESULTS [IN AUC] OF BEMTL'S VARIANTS.

	A_WClothes		A_Electronics	
Model	**Sentiment**	**Recom**	**Sentiment**	**Recom**
BeSTL	84.21%	86.07%	84.59%	83.00%
Vanilla-BeMTL	84.96%	87.65%	86.23%	83.22%
BeMTL-CNN	84.11%	86.86%	84.69%	82.21%
BeMTL	**85.59%**	**88.19%**	**86.78%**	**83.22%**

2) Comparison with Multi-Task Learning (MTL) Methods: In comparison with the MTLFS, MMAM, and MTLSS state-of-the-art MTL methods for sentiment analysis, the experimental results in Table II indicate that BeMTL outweighs them with a significant performance on the two datasets across the two tasks. Beyond other settings, the good performance of BeMTL over these methods is attributed to the contextualized embeddings produced by pre-trained BERT model, the complementary of contextual features extracted by CNN and MHA. It is also because of exploiting the correlation information between sentiment and explicit recommendation classification tasks. Moreover, compared with the STL approaches, the results achieved by these MTL methods evidence the potential of MTL setting on these kinds of problems.

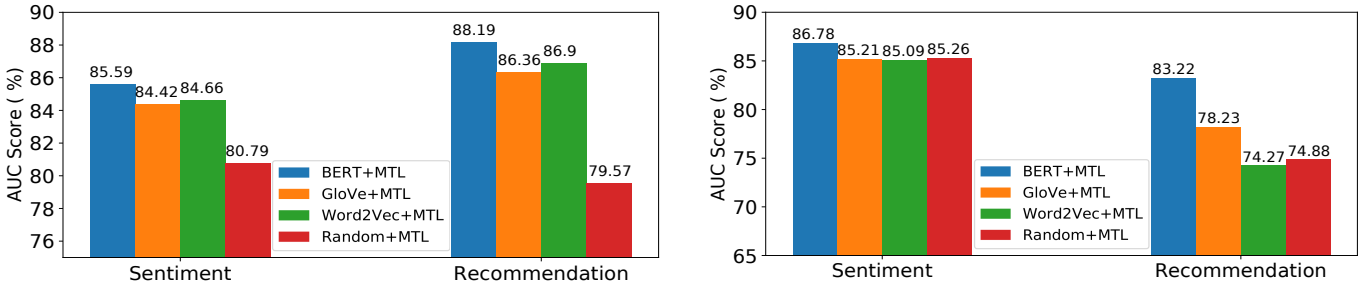
E. Ablation Study

In this section, we ran some ablation experiments to illustrate the contributions of the BeMTL's components. The experimental results achieved by BeMTL variants are shown in Table III (Recom means explicit recommendation).

Effect multi-task learning. To evaluate the power of MTL, we compare BeMTL and BERT-enhanced Single-task learning (BeSTL). BeSTL is our standalone classifier separately trained on an individual task. BeSTL applies the softmax on the representation S_{rep} or R_{rep} for sentiment or explicit recommendation classification. In view of the results in Table III, we observe that BeMTL outperforms BeSTL for sentiment and explicit recommendation classification tasks across both datasets. Thus, the results validate our initial assumption that when our two tasks are jointly trained, the knowledge in each task can help to improve the classification performance.

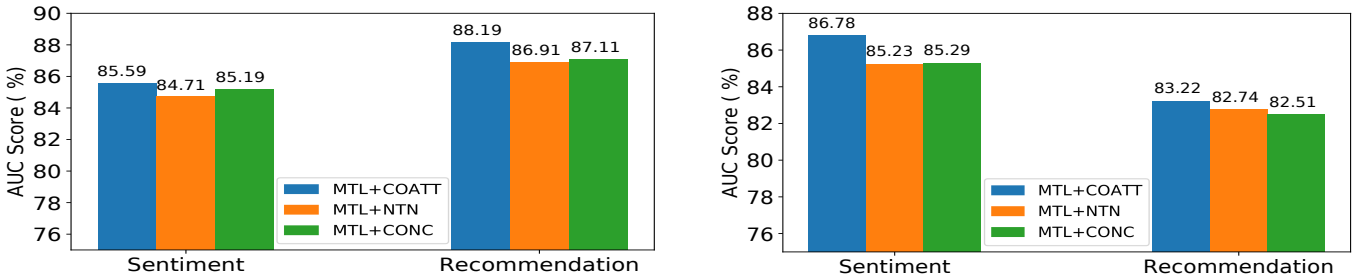
Effect of the inter-task matching layer. To assess the advantage offered by the inter-task matching layer (Subsection III-F), we compare BeMTL with Vanilla-MTL. Vanilla-BeMTL does contain the IML component. It simultaneously performs sentiment and explicit recommendation classification S_{rep} and R_{rep}. From the results in Table III, we observe that BeMTL is superior to Vanilla-BeMTL on the two datasets. Thus, the results prove our initial idea that exploiting the relationship between related tasks can help to improve the results.

Effect of contextual features' complement. To gauge the significance of utilizing both types of contextual features, i.e., local contextual features and global contextual features, we compare BeMTL against BeMTL-CNN. BeMTL-CNN is a BeMTL's variant, which does not contain a multi-head attention block. Instead, it performs sentiment analysis and explicit



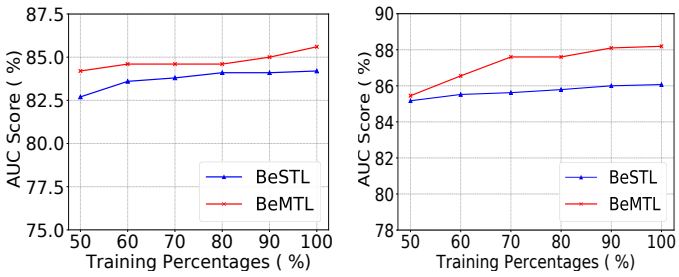
(a) A_WClothes dataset (b) A_Electronics dataset

Fig. 3. Performance comparison of our MTL with respect to word embedding methods.



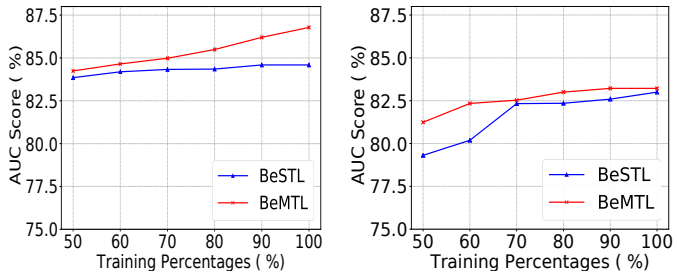
(a) A_WClothes dataset (b) A_Electronics dataset

Fig. 4. Performance comparison of our MTL with respect to the fusion methods.



(a) Sentiment (b) Recommendation

Fig. 5. Performance comparison of BeMTL and BeSTL with respect to the training percentage on A_WClothes dataset.



(a) Sentiment (b) Recommendation

Fig. 6. Performance comparison of BeMTL and BeSTL with respect to the training percentage on A_Electronics dataset.

recommendation classification based only on the shared local contextual features extracted by CNN and the rest is the same as the full BeMTL model. The results in Table III indicate that the omission of the multi-head attention block in BeMTL-CNN causes the performance to drop for both tasks across the two datasets. Therefore, the results confirm that exploiting both local and global contextual features is of great advantage.

Effect of pre-trained BERT model. To evaluate the contribution of pre-trained BERT as a powerful word embedding method, we perform an ablation study by comparing the performance of our MTL with respect to various word embedding methods. Those methods include GloVe [30], word2vec [31] and, random word embedding methods. To have a fair comparison with pre-trained BERT, we did not fine-tune the produced embeddings from all compared methods. As is expected, the pre-trained BERT contextual embedding method has a noticeable effect on the results, as is shown in Fig. 3, where our MTL initialized with pre-trained BERT has the best performance. Therefore, the results are in accordance with the study [5], which has suggested the use of pre-trained BERT as a way to improve the results. However, it should be noted that without pre-trained BERT, our MTL, which is initialized with word2vec and GloVe on A_WClothes and A_Electronics datasets respectively, still outperforms all baseline approaches.

F. Parameter Sensitivity Analysis

Effect of fusion methods. To investigate the influence of fusion method used in the inter-task matching layer (Subsection III-F) on BeMTL's performance, we conduct experiments and compare the performance of our model with coattention against other fusion methods like neural tensor network (NTN) and concatenation methods. We present the results in Fig. 4. As is observed, our BeMTL model fares best when the coattention

is used. Thus, the results prove the capability of coattention in matching the necessary features.

Effect of training set size. We study how the training set size affects the performance of BeMTL and BeSTL models. In Fig. 5 and Fig. 6, we vary the training percentage of our samples in our datasets. Foremost, it can be observed that the performance of both models increases with the training ratio. In comparison with BeSTL, BeMTL can still yield better performance even with a small proportion of training samples. Thus, the results well prove its generalization capability.

V. Conclusion

In this paper, we introduce a BERT-enhanced multi-task learning model (BeMTL) approach to improve sentiment analysis by using an MTL of explicit recommendation classification task. The proposed BeMTL takes the embeddings produced by the pre-trained BERT-based embedding layer and then applies convolutional multi-head attention to model shared sentence contextual representations. To adequately capture the correlation information between both tasks, the inter-task matching layer (IML) is applied to generate matching representations, which are combined with the task-specific features. The proposed BeMTL effectively improves the performance of our main task, i.e, sentiment analysis as well as the auxiliary task, i.e, explicit recommendation on two publicly available datasets. Particularly, it consistently outperforms state-of-the-art methods in sentiment analysis. Thus, this work validates our initial hypothesis that jointly training the sentiment analysis and explicit recommendation classification tasks can help to improve the performance.

As explicit recommendation is proved to be a new hidden signal that can help to improve the performance of sentiment analysis, in future work, we intend to build new large datasets for rigorous experiments. Furthermore, we will combine the implicit recommendations generated by the recommendator system with sentiment and explicit recommendation for the better explainable recommendation.

Acknowledgement

This work is supported by the National Key Research and Development Program of China under grants 2016QY01W0202 and 2016YFB0800402, National Natural Science Foundation of China under grants U1936108, U1836204, 61572221, 61572222, and 61502185.

References

[1] G. Packard and J. Berger, "How language shapes word of mouth's impact," *Journal of Marketing Research*, vol. 54, pp. 572–588, 2017.

[2] B. Liu, *Sentiment Analysis - Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2015.

[3] A. F. Agarap and P. Grafilon, "Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network (RNN)," *CoRR*, vol. abs/1805.03687, 2018.

[4] G. João and R. Paulo, "How to predict explicit recommendations in online reviews using text mining and sentiment analysis," *Journal of Hospitality and Tourism Management*, pp. 1–4, 2019.

[5] O. Habimana, Y. Li, R. Li, X. Gu, and G. Yu, "Sentiment analysis using deep learning approaches: An overview," *Science China Information Sciences*, vol. 63, no. 1:111102, 2020.

[6] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1:41–75, 1997.

[7] A. Maurer, M. Pontil, and B. Romera-Paredes, "The benefit of multitask representation learning," *J. Mach. Learn. Res.*, vol. 17, pp. 1–32, 2016.

[8] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, and A. F. Gelbukh, "Sentiment and sarcasm classification with multitask learning," *IEEE Intelligent Systems*, vol. 34, no. 3, pp. 38–43, 2019.

[9] H. Zhang, L. Xiao, W. Chen, Y. Wang, and Y. Jin, "Multi-task label embedding for text classification," in *Proceedings of the Conference on EMNLP*, 2018, pp. 4545–4553.

[10] G. Balikas, S. Moura, and M. Amini, "Multitask learning for fine-grained twitter sentiment analysis," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 1005–1008.

[11] Z. Dai, W. Dai, Z. Liu, F. Rao, H. Chen, G. Zhang, Y. Ding, and J. Liu, "Multi-task multi-head attention memory network for fine-grained sentiment analysis," in *Natural Language Processing and Chinese Computing - 8th CCF International Conference*, 2019, pp. 609–620.

[12] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit, "One model to learn them all," *CoRR*, vol. abs/1706.05137, 2017.

[13] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the NAACL–HLT*, 2019, pp. 4171–4186.

[14] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1746–1751.

[15] Y. Zhang and B. C. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017, pp. 253–263.

[16] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proceedings of the ACL*, 2017, pp. 562–570.

[17] H. Schwenk, L. Barrault, A. Conneau, and Y. LeCun, "Very deep convolutional networks for text classification," in *Proceedings of the EACL*, 2017, pp. 1107–1116.

[18] A. E. Mousa and B. W. Schuller, "Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis," in *Proceeding of the EACL*, 2017, pp. 1023–1032.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations*, 2015.

[20] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, "Hierarchical attention networks for document classification," in *The Conf. of the NAACL–HLT*, 2016, pp. 1480–1489.

[21] Z. Wu, X. Dai, C. Yin, S. Huang, and J. Chen, "Improving review representations with user attention and product attention for sentiment classification," in *Proceedings of AAAI*, 2018, pp. 5989–5996.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[23] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, "Disan: Directional self-attention network for rnn/cnn-free language understanding," in *Proceedings of AAAI*, 2018, pp. 5446–5455.

[24] N. Wang, H. Wang, Y. Jia, and Y. Yin, "Explainable recommendation via multi-task learning in opinionated text data," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 165–174.

[25] Y. Lu, R. Dong, and B. Smyth, "Why I like it: multi-task learning for recommendation and explanation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 4–12.

[26] A. W. Yu, D. Dohan, M. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," in *ICLR*, 2018.

[27] B. Yang, L. Wang, D. F. Wong, L. S. Chao, and Z. Tu, "Convolutional self-attention networks," in *Proceedings of NAACL–HLT*, 2019, pp. 4040–4045.

[28] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," in *ICLR*, 2017.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, 2015.

[30] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. of the EMNLP*, 2014, pp. 1532–1543.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR*, 2013.