Multi-Task Neural Learning Architecture for End-to-End Identification of Helpful Reviews

Miao Fan[1,2,3], Yue Feng[1,2,3], Mingming Sun[1,2,3], Ping Li[1,2], Haifeng Wang[1,3], Jianmin Wang[4]
[1]Baidu Research
[2]Big Data Lab (BDL-US), Baidu Research
[3]National Engineering Laboratory of Deep Learning Technology and Application, China
[4]School of Software Engineering, Tsinghua University
[1]{fanmiao,fengyue04,sunmingming01,liping11,wanghaifeng}@baidu.com
[4]jimwang@mail.tsinghua.edu.cn

***Abstract*—Helpful reviews play a pivotal role in recommending desirable goods and accelerating purchase decisions of customers in e-commercial services. Given a large proportion of product reviews with unknown helpfulness/unhelpfulness, the research on automatic identification of helpful reviews has drawn much attention in recent years. However, state-of-the-art approaches still rely heavily on extracting heuristic text features from reviews with domain-specific knowledge. In this paper, we first introduce a multi-task neural learning (MTNL) architecture for identifying helpful reviews. The end-to-end neural architecture can learn to reconstruct effective features upon the raw input of words and even characters, and the multi-task learning paradigm helps to make more accurate predictions of helpful reviews based on a secondary task which fits the star ratings of reviews. We also build two datasets containing helpful/unhelpful reviews from different product categories in Amazon, and compare the performance of MTNL with several mainstream methods on both datasets. Experimental results confirm that MTNL outperforms the state-of-the-art approaches by a significant margin.**

***Index Terms*—Helpful review identification, E-commerce, Multi-task learning, Deep neural networks, Attention mechanism**

I. Introduction

Customer reviews are ubiquitously available online for a wide range of products and services on the web sites such as Amazon[1] and Yelp[2]. Among the numerous reviews, the helpful ones play a pivotal role in recommending desirable goods and accelerating customers' purchase decisions, as helpful reviews are more likely to provide supplementary and convincing features of a product from the buyers in addition to its posted descriptions.

Given the fact that a product/service may be commented by hundreds or thousands of buyers online with uneven quality, it is scarcely possible for a potential customer to go through all the reviews to find the helpful ones. Therefore, several e-commercial web sites, e.g., Amazon.com, deploy a crowd-sourcing module, as shown by Figure 1, which encourages users to collaboratively vote for the helpfulness of each review,

Fig. 1: An example of a product which has 586 reviews in Amazon.com.

and the most helpful reviews will have the priority of being displayed to all visitors. This feature proven by a recent study increases the revenue of Amazon.com with an estimated **27 billion** U.S. dollars annually.[3]

This module "Was this review helpful to you?" (see Figure

[1]https://www.amazon.com/
[2]https://www.yelp.com/

IEEE/ACM ASONAM 2018, August 28-31, 2018, Barcelona, Spain
978-1-5386-6051-5/18/$31.00

[3]http://www.uie.com/articles/magicbehindamazon.

1) on Amazon.com alleviates the demand of discovering helpful reviews to a certain extent. Nevertheless, it is far from satisfying the emerging need for automatically identifying helpful reviews, because roughly **54.85%** product reviews in Amazon.com do NOT receive any vote of helpfulness/unhelpfulness[4]. The phenomenon on a large proportion of reviews with unknown helpfulness is even significant in low-traffic items such as those less popular and/or new arrival products.

To address the critical issue, a series of work in recent years has dedicated to the research on automatically assessing the helpfulness of a review based on its various features. To the best of our knowledge, they are virtually all human-defined with domain-specific knowledge, including structural [3], lexical [4, 5], emotional [6], and even semantic features [7, 8]. These features will generally be fed into an off-the-shelf SVM classifier [9, 10] for identification. We, however, believe that the techniques of feature engineering request tedious labor, and the human-annotated corpora are not sufficient for training practicable systems to identify helpful reviews.

The recent emergence of deep learning [11, 12, 13] provides an insight on using deep neural networks (DNNs) [14, 15, 16] to reduce the dependence on domain-specific knowledge and feature engineering. In the meanwhile, several applications leveraging the multi-task learning paradigm [17, 18, 19] have verified that the auxiliary task can help to train the under-layer embeddings more sufficiently and to prevent the entire model from over-fitting the main task. Moreover, we find out that the star rating [20, 21] of a review is an option as the auxiliary task, as *nearly 54% helpful reviews have extreme (i.e., 1-star and 5-star) ratings* where strong but useful opinions tend to be expressed.

Inpsired by the aforementioned prior work, in this paper we first introduce a multi-task neural learning (MTNL) architecture to identify the helpfulness of a product review in an end-to-end fashion. Hhe proposed MTNL architecture makes contributions from two perspectives: 1) It can directly take the words and even characters from review texts as raw inputs of MTNL without resorting to techniques of feature extraction under domain-specific knowledge. In our framework, effective features can be automatically reconstructed along with adapting the entire model to achieving the main task: identification of helpful reviews. 2) The multi-task learning paradigm provides an auxiliary and correlated target: star rating regression of reviews, which helps to train the under-layer word/character -level embeddings more sufficiently and to make more accurate predictions of helpful reviews.

To further assist us to study this problem, we also construct two datasets from different product categories: Clothes and Electronics in Amazon. Both contain helpful and unhelpful reviews collaboratively voted by the users online. To demonstrate the effectiveness of MTNL, we compare the performance of our approach with several mainstream methods on both datasets[5]. Experimental results confirm that MTNL outperforms the state-of-the-art approach with averaged improvements of **2.42%** in accuracy, **1.48%** in F1 score and **2.85%** in AUC [22] with p-value < 0.05 in the significance testing (t-test) [23, 24]. Extensive studies are further conducted to discuss the effects of character-level embedding, attention mechanism and multi-task learning paradigm employed by MTNL, and more empirical results show that these key components contribute to the state-of-the-art performance of MTNL to varying degrees, not only within domain-specific but also on cross-domain datasets.

II. Related Work

To the best of our knowledge, the prior work on identification of helpful reviews virually all relies heavily on (manually) extracting various features of review texts. In line with [8] and [7], the features used in prior work include the following:

- Structural features [3, 25]: The star rating of a review is a structural but important evidence, as it indicates the opinion of the buyers on the product. Besides that, the number of tokens, the number of sentences, and the average length of sentences are all considered as the structural features of a review.
- Lexical features [4, 5]: After removing stop words and non-frequent words ($tf < 3$), unigrams and bigrams are extracted and weighted by the measurement of *tf-idf* [26] as the lexical features.
- Syntactic features [4]: If we go deeper to parse a review, we can obtain the part-of-speech (POS) tag of each token. The syntactic features are composed of the percentages of tokens that are nouns, verbs, adjectives, and adverbs, respectively.
- Emotional features [6]: The Geneva Affect Label Coder (GALC) dictionary [27] defines 36 emotion states which are distinguished by words. The emotional features contain the number of occurrences of each emotional word plus one additional dimension for the number of non-emotional words.
- Semantic features [7]: The General Inquirer (INQUIRER) dictionary [28] can help to map each word into a semantic tag. Similar to the emotional features, the semantic features are organized by a vector which records the number of occurrences of each semantic tag.
- Argument features [8]: The evidence-conclusion discourse relations, also known as arguments, are more intricate linguistic features of reviews. Different granularities of argument features, e.g., the number of arguments, the number of words in arguments and etc., are included.

After the features are (manually) constructed, an off-the-shelf classifier such as SVM [9, 10] Random Forest [29, 30], or gradient boosting [31, 32] is then adopted to learn with those features to identify the helpfulness of a review.

[4]We have checked all the data [1, 2], i.e. 82.68 million reviews from 24 product categories in Amazon.com, in http://jmcauley.ucsd.edu/data/amazon/links.html.

[5]These two datasets are available at https://drive.google.com/drive/folders/1fAPa4LzogYZOppSd2fpVS10vbId47ZMM?usp=sharing.

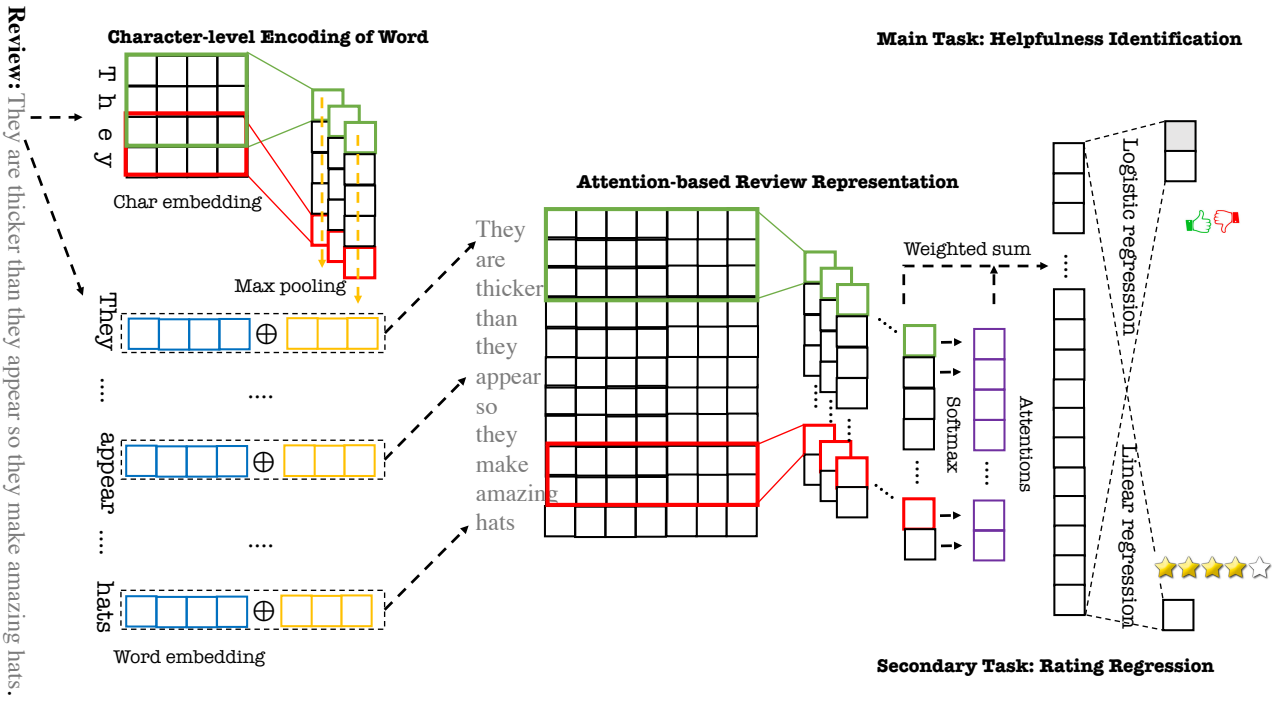

Fig. 2: Illustration of the proposed end-to-end architecture of multi-task neural learning (MTNL) as the solution to the problem of identification of helpful reviews.

III. Learning Architecture

In this section, we introduce our end-to-end architecture of multi-task neural learning (MTNL) to tackle the challenging task of helpful review identification. As illustrated by Figure 2, MTNL takes the raw characters of a review as input and encodes a fix-dimension character-level embedding for each word through a single layer convolutional neural network (CNN) [16, 33]. We then concatenate each character-level encoding [34] to its corresponding word embedding within a review. These joint embeddings are fed into another CNN equipped with the attention mechanism [35, 36] to produce the distributed representation of the review. The attention-based review representation records the reconstructed features from the raw inputs (i.e., words and characters), and will simultaneously adapt itself to the primary and secondary learning targets, i.e., helpfulness identification and rating regression of reviews, respectively.

A. Character-Level Encoding of Word

Given the fact that each review is a sequence of words, we define that a review which has m words is represented as a list $v_{1:m}$:

$$v_{1:m} = [v_1, v_2, ..., v_m], \tag{1}$$

where $v_{i:j}$ refers to a subsequence containing the words from the i-th index to the j-th index of the review. We use the bold font $\mathbf{v}_i \in \mathbb{R}^d$ to denote the word-level embedding of v_i with the dimension of d. By default, the embeddings are viewed as column vectors in this paper.

Suppose that the i-th word v_i is composed of a sequence of n characters denoted by $u_{1:n}$:

$$u_{1:n} = [u_1, u_2, ..., u_n]. \tag{2}$$

The vector representations of the characters are:

$$\mathbf{u}_{1:n} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_n]. \tag{3}$$

Here we use d' to denote the dimension of character-level embeddings: $\mathbf{u}_j \in \mathbb{R}^{d'}$.

With the aid of narrow convolution [37], each filter can produce distributed representations of character n-grams exactly with the same length. For instance, if we apply a filter $\mathbf{w}_i \in \mathbb{R}^{d' \times l'}$ which denotes the i-th filter with a window of l' characters, to a sequence of characters $\mathbf{u}_{j:j+l'-1} \in \mathbb{R}^{d' \times l'}$, a convoluted feature $p_{i,j}$ of character l'-grams is produced by:

$$p_{i,j} = \tanh(\mathbf{w}'_i \cdot \mathbf{u}_{j:j+l'-1} + b'), \tag{4}$$

where $b' \in \mathbb{R}$ is a bias term corresponding to the vector of k' filters $\mathbf{W}' = [\mathbf{w}'_1, \mathbf{w}'_2, ..., \mathbf{w}'_{k'}]$. The feature map $\mathbf{P} = \left[\mathbf{p}_1^{\mathrm{T}}, \mathbf{p}_2^{\mathrm{T}}, ..., \mathbf{p}_{k'}^{\mathrm{T}}\right]^{\mathrm{T}}$ encodes k' kinds of l'-gram character embeddings of the word. In order to obtain the fix-dimension character-level encoding of the word v_i, we apply the max-pooling strategy to each row of $\mathbf{P}$:

$$\mathbf{h} = \left[\max(\mathbf{p}_1^{\mathrm{T}}), \max(\mathbf{p}_2^{\mathrm{T}}), ..., \max(\mathbf{p}_{k'}^{\mathrm{T}})\right]^{\mathrm{T}}, \tag{5}$$

where $\mathbf{h}_i \in \mathbb{R}^{k'}$ is the k'-dimension encoding of the word v_i regardless of its length l'.

After obtaining both word-level (e.g., $\mathbf{v}_i \in \mathbb{R}^d$) and character-level (e.g., $\mathbf{h}_i \in \mathbb{R}^{k'}$) encodings of the words, we concatenate each pair of them, i.e. $\mathbf{e}_i = \mathbf{v}_i \oplus \mathbf{h}_i$, to generate the underlying distributed representations of the review:

$$\mathbf{e}_{1:m} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m], \tag{6}$$

where $\mathbf{e}_i \in \mathbb{R}^{d+k'}$.

B. Attention-Based Review Representation

After we obtain the joint vector representations of the words, e.g., $\mathbf{e}_i \in \mathbb{R}^{d+k'}$ in a review, we apply another CNN layer to $\mathbf{e}_{1:m}$, similar with Eq. (4), to generate the convoluted feature map $\mathbf{Q}$ of l-grams where an entry is produced by

$$q_{i,j} = \tanh(\mathbf{w}_i \cdot \mathbf{e}_{j:j+l-1} + b), \tag{7}$$

where $b \in \mathbb{R}$ is another bias term corresponding to the vector of k filters $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_k]$. Different from $\mathbf{P}$, we consider the feature map $\mathbf{Q}$ from another perspective that $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_{m-l+1}]$. Each column of $\mathbf{Q}$ encodes the embedding of an l-gram with length of k. Then a weight a_i is computed by a single-time attention model for each $\mathbf{q}_i \in \mathbb{R}^k$ as follows:

$$a_i = \text{softmax}(\mathbf{r}^{\mathrm{T}} \tanh(\mathbf{W}_a \mathbf{q}_i)), \tag{8}$$

where $\mathbf{r}$ and $\mathbf{W}_a$ are the parameters of the single-time attention model to be learnt. Specifically, $\mathbf{r} \in \mathbb{R}^k$ and $\mathbf{W}_a \in \mathbb{R}^{k \times k}$.

Finally, the attention-based representation of the review denoted by $\mathbf{x}$ is the weighted sum over $\mathbf{q}_i$ in terms of a_i:

$$\mathbf{x} = \sum_{i=1}^{m-l+1} a_i \mathbf{q}_i. \tag{9}$$

C. Secondary Task: Rating Regression

The attention-based distributed review representation, i.e., $\mathbf{x}$, is expected to fit the star rating of the corresponding review as the secondary learning target of MTNL. Specifically, we use a linear model to predict the rating $\hat{z}$ of the review:

$$\hat{z} = \mathbf{w}_z^{\mathrm{T}} \mathbf{x} + b_z, \tag{10}$$

where $\mathbf{w}_z \in \mathbb{R}^k$ is the weight and the scalar b_z is the bias with respect to the linear model. Given a collection of N training samples with ground truths of ratings $(z_1, z_2, ..., z_N)$, we define the loss of regression on the secondary task as follows,

$$\mathcal{L}_z = \sum_{i=1}^{N} (z_i - \hat{z}_i)^2. \tag{11}$$

The problem of rating predictions has been extensively studied in the literature; see for example the recent work [38].

D. Primary Task: Helpfulness Identification

Our primary task aims at leveraging the neural representation of a review, i.e., $\mathbf{x}$, in order to judge whether the review is helpful. This is a binary classification problem and the logistic regression model is conveniently adopted:

$$\hat{y} = \frac{1}{1 + \exp^{-(\mathbf{w}_y^{\mathrm{T}} \mathbf{x} + b_y)}}, \tag{12}$$

where $\mathbf{w}_y \in \mathbb{R}^k$ is the weight and the scalar b_y is the bias with respect to the classification model. Given the same set of N training samples with ground truths of helpfulness $(y_1, y_2, ..., y_N)$, we define the loss of classification on the primary task as follows,

$$\mathcal{L}_y = \sum_{i=1}^{N} (y_i - 1) \log(1 - \hat{y}_i) - y_i \log(\hat{y}_i). \tag{13}$$

E. Combining Both Tasks

Overall, MTNL learns from the N training samples to minimize the joint loss of $\mathcal{L}_y$ and $\mathcal{L}_z$ with a trade-off hyper-parameter $\lambda \in [0.0,\ 1.0]$:

$$\mathcal{L} = \lambda \mathcal{L}_y + (1 - \lambda) \mathcal{L}_z. \tag{14}$$

IV. Experiments

A. Datasets

We build two datasets containing helpful and unhelpful reviews from different product categories in Amazon: Clothes and Electronics, where the helpfulness of each review is collaboratively voted by customers. Specifically, we regard a review which receives more than 50% votes of helpfulness as a helpful one, and less than 50% votes of helpfulness as an unhelpful one.

TABLE I: Statistics of the Clothes and Electronics datasets constructed from Amazon.com.

Category	Item	Subset	
		Train	*Test*
Clothes	#(Review)	20,000	4,000
	– #(Helpful)	10,000	2,000
	– #(Unhelpful)	10,000	2,000
	#(Sentence)	94,202	19,166
	#(Vocabulary)	38,207	14,803
	#(Avg. L/Sen)	17.59	17.63
Electronics	#(Review)	20,000	4,000
	– #(Helpful)	10,000	2,000
	– #(Unhelpful)	10,000	2,000
	#(Sentence)	148,243	29,599
	#(Vocabulary)	85,310	29,144
	#(Avg. L/Sen)	22.66	22.77

As shown by Table I, each dataset contains 20,000 reviews for training and 4,000 reviews for testing. Both datasets are balanced, i.e., half reviews are helpful and the others are unhelpful. On the hand, more statistics, such as the number of

sentences, the size of vocabulary and the average length per sentence, indicate that the two datasets from various domains are quite different.

We explore the identical vocabulary words that the two datasets share in addition, and find out that they only have about 17,000 words (44.49% in Clothes and 19.92% in Electronics) in common. Moreover, there are 30.95% out-of-vocabulary (OOV) words [39] in the test set of Clothes and 39.08% OOV words in the test set of Electronics. The large proportion of OOV words challenges the generalization ability of models as well.

B. Evaluation Setups

Experiments are conducted on the two datasets by comparing MTNL with several mainstream approaches on helpful review identification, including STR (structural features) + SVM [3], UGR/BGR (lexical features) + SVM [4, 5], SYN (syntactic features) + SVM [4], GALC (emotional features) + SVM [6], and INQUIRER (semantic features) + SVM [7]. Besides those approaches with their specific category of features, we further concatenate GALC with INQUIRER to generate the features of Fusion$_{sem}$. And Fusion$_{all}$ contains all the extracted features mentioned above that encode various types of domain knowledge. The argument features exploited by [8] are hardly reproducible, as the intricate features they used require a large scale of annotated corpus to build an argument extraction system which is not available for now.

As the main task is a classification problem, we adopt F1, accuracy and AUC (the Area Under an ROC Curve) to measure the performance of all the models. The best combinations of the hyper-parameters for the two datasets are selected by the five-fold cross-validation on respective training sets. Here we describe the hyper-parameters of MTNL tried on the two datasets: trials on the dimension of character embedding $d' = \{16, 32, 64, 128\}$, the number of filters for the character-level encoding $k' = \{16, 32, 64\}$, the window sizes of the filters for the character-level encoding $l' = \{1, 2, 3, 4\}$, the dimension of word embedding $d = \{20, 50, 100, 200\}$, the number of filters for the attention-based review representation $k = \{16, 32, 64\}$, the window sizes of the filters for the review representation $l = \{1, 2, 3, 4\}$ and the trade-off hyper-parameter $\lambda = \{0.2, 0.4, 0.5, 0.6, 0.8\}$. Finally, we choose the model of MTNL with $d' = 32, d = 50, k' = 32, k = 32, l' = \{1, 2, 3, 4\}, l = \{1, 2, 3, 4\}$ and $\lambda = 0.5$ on the Clothes dataset, and the model of MTNL with $d' = 64, d = 100, k' = 16, k = 32, l' = \{1, 2, 3, 4\}, l = \{1, 2, 3, 4\}$ and $\lambda = 0.5$ on the Electronics dataset.

C. Comparison Results

Table II and Table III display the comparison results on the Amazon Clothes and Amazon Electronics datasets, respectively. Those results demonstrate that 1) UGR + SVM achieves the state-of-the-art F1 score and accuracy among the prior arts; 2) Fusion$_{all}$ + SVM obtains the highest score of AUC among the mainstream approaches. The experimental results on the prior mainstream approaches are in accordance with the results reported by [8]. Furthermore, 3) MTNL consistently outperforms the state-of-the-art approach on both datasets with averaged improvements of 2.42% in accuracy, 1.48% in F1 and 2.85% in AUC. We also conduct significant testing (t-test) on the improvements of MTNL over the best baselines, i.e., UGR+SVM and Fusion$_{all}$+SVM, and the p-value < 0.05 indicates that the improvements measured by accuracy, F1 and AUC (denoted with '*' in Table II and Table III) are significant on both datasets.

V. Discussions

In this section, we plan to discuss the effectiveness of several key components in the end-to-end MTNL, i.e. CNN for character-level embedding, the attention mechanism for distributed representations of reviews, and the multi-task learning paradigm for helpful review identification, respectively. Then the generalization ability of the whole MTNL architecture is assessed by cross-domain evaluations with the two datasets: Amazon Clothes and Amazon Electronics.

A. Effects of Character-level Embedding

The module of CNN for character-level encoding can generate distributed representations in a fixed dimension for different words with variable lengths. Moreover, the out-of-vocabulary (OOV) issue which widely exists in traditional approaches based on heuristic feature engineering will be alleviated by the character-level embedding, as the OOV words may share the same root or affix structures with the words in the vocabulary to generate similar embeddings. We take turns to turn off the modules of character- and word- level embedding in STNL and MTNL, and test their performance on the Clothes and Electronics datasets. Table IV reports the results of the experiments which demonstrate that the module of character-level embedding brings improvements of 1.24% in F1, 1.76% in accuracy and 1.29% in AUC for STNL, and improvements of 0.74% in F1, 0.96% in accuracy and 1.03% in AUC for MTNL. Even without any embeddings of words, the MTNL (w/o word embeddings) model still achieves comparable results with the mainstream approaches on both datasets.

B. Effects of Attention Mechanism

A straight-forward approach of aggregating the multi-granular embeddings, i.e., the feature map $\mathbf{Q}$ produced by the CNN (see Eq. (7)) for a review, is to do average/max pooling [40] on them to generate the review representation. However, we consider the average-pooling layer [41] tends to wipe out the significant features, and the max-pooling layer [42] makes the effective features untraceable. The attention mechanism we adopt, re-weights the multi-granular embeddings to represent a review, which highlights the effective multi-granular features and makes them interpretable. We conduct experiments on replacing the attention module with average and max pooling in STNL and MTNL on the two datasets. The results of STNL/MTNL (w/o attention mechanism) in Table IV are the mean performance of leveraging average and max pooling.

TABLE II: Comparison results on the Amazon Clothes dataset.

MODEL	F1	Precision	Recall	Accuracy	AUC
STR + SVM [3]	47.35	58.54	39.75	55.80	58.40
UGR + SVM [5]	*58.87*	56.88	61.00	*57.38*	60.22
BGR + SVM [4]	57.19	56.58	57.80	56.73	59.70
SYN + SVM [4]	54.43	52.25	56.80	52.45	53.05
GALC + SVM [6]	46.67	57.65	39.20	55.20	57.82
INQUIRER + SVM [7]	49.62	57.64	43.55	55.78	60.30
Fusion$_{sem}$ + SVM [7]	46.88	58.09	39.30	55.47	58.46
Fusion$_{all}$ + SVM [7]	48.40	59.18	40.95	56.35	*60.71*
Fusion$_{sem}$ + Random Forest	50.10	56.98	44.70	55.48	56.15
Fusion$_{all}$ + Random Forest	52.02	56.43	48.25	55.50	59.15
MTNL	$\mathbf{61.11}^*$	59.42	62.90	$\mathbf{59.98}^*$	$\mathbf{63.78}^*$

TABLE III: Comparison results on the Amazon Electronics dataset.

MODEL	F1	Precision	Recall	Accuracy	AUC
STR + SVM [3]	51.63	65.05	42.80	59.90	64.43
UGR + SVM [5]	*62.86*	62.18	63.55	*62.45*	66.19
BGR + SVM [4]	61.37	60.91	61.85	61.08	65.75
SYN + SVM [4]	53.98	52.72	55.30	52.85	53.76
GALC + SVM [6]	52.31	65.93	43.35	60.48	64.81
INQUIRER + SVM [7]	53.49	66.69	44.65	61.18	65.74
Fusion$_{sem}$ + SVM [7]	51.69	65.00	42.90	59.90	64.52
Fusion$_{all}$ + SVM [7]	53.61	66.84	44.75	61.28	*66.67*
Fusion$_{sem}$ + Random Forest	56.25	60.72	52.40	59.25	61.69
Fusion$_{all}$ + Random Forest	54.98	62.72	48.95	59.93	63.79
MTNL	$\mathbf{63.57}^*$	65.62	61.65	$\mathbf{64.68}^*$	$\mathbf{69.29}^*$

TABLE IV: Effects of character/word-level embedding and attention mechanism in STNL and MTNL on the Amazon Clothes and Amazon Electronics datasets.

CATEGORY	MODEL	F1	Precision	Recall	Accuracy	AUC
Clothes	STNL (w/o char embeddings)	56.98	57.68	56.30	57.50	60.04
	STNL (w/o word embeddings)	53.66	56.32	51.25	55.75	56.87
	STNL (w/o attention mechanism)	54.64	56.97	52.50	56.43	58.18
	STNL	59.26	57.89	60.70	58.28	60.41
	STNL (with star ratings)	60.94	58.80	63.25	59.02	62.43
	MTNL (w/o char embeddings)	60.76	58.46	63.25	59.15	62.85
	MTNL (w/o word embeddings)	60.53	57.46	63.95	58.30	60.50
	MTNL (w/o attention mechanism)	60.77	57.53	64.40	58.43	60.93
	MTNL	**61.11**	59.42	62.90	**59.98**	**63.78**
Electronics	STNL (w/o char embeddings)	62.02	58.65	65.80	59.70	62.73
	STNL (w/o word embeddings)	58.91	58.68	59.15	58.75	61.47
	STNL (w/o attention mechanism)	59.23	58.10	60.40	58.43	61.05
	STNL	62.21	62.57	61.85	62.43	65.94
	STNL (with star ratings)	62.67	64.23	61.20	63.80	67.95
	MTNL (w/o char embeddings)	62.45	64.48	60.55	63.60	68.17
	MTNL (w/o word embeddings)	60.80	58.54	63.25	59.23	61.81
	MTNL (w/o attention mechanism)	61.35	59.17	63.70	59.88	62.55
	MTNL	**63.57**	65.62	61.65	**64.68**	**69.29**

TABLE V: Experimental results for comparing the ability to cross-domain generalization among MTNL and several advanced approaches, i.e., UGR + SVM and Fusion$_{all}$ + SVM.

MODEL	F1	Precision	Recall	Accuracy	AUC
UGR + SVM (Train: *Electronics*, Test: Clothes)	45.10	59.01	36.50	55.58	59.95
UGR + SVM (Train: Clothes, Test: Clothes)	58.87	56.88	61.00	57.38	60.22
Fusion$_{all}$ + SVM (Train: *Electronics*, Test: Clothes)	24.90	64.98	15.40	53.55	56.73
Fusion$_{all}$ + SVM (Train: Clothes, Test: Clothes)	48.40	59.18	40.95	56.35	60.71
MTNL (Train: *Electronics*, Test: Clothes)	53.08	63.79	45.45	59.83	63.77
MTNL (Train: Clothes, Test: Clothes)	61.11	59.42	62.90	59.98	63.78
UGR + SVM (Train: *Clothes*, Test: Electronics)	55.95	61.32	51.45	59.50	63.39
UGR + SVM (Train: Electronics, Test: Electronics)	62.86	62.18	63.55	62.45	66.19
Fusion$_{all}$ + SVM (Train: *Clothes*, Test: Electronics)	49.23	58.58	42.45	60.03	64.47
Fusion$_{all}$ + SVM (Train: Electronics, Test: Electronics)	53.61	66.84	44.75	61.28	66.67
MTNL (Train: *Clothes*, Test: Electronics)	61.79	63.69	60.00	62.90	67.14
MTNL (Train: Electronics, Test: Electronics)	63.57	65.62	61.65	64.68	69.29

We can tell that the attention mechanism brings about a leap forward in performance with improvements of 3.80% in F1, 2.93% in accuracy and 3.56% in AUC in STNL, and improvements of 1.28% in F1, 3.18% in accuracy and 4.80% in AUC in MTNL.

C. Multi-task vs. Single-task Learning

Neural learning architectures often enjoy the benefits of adapting feature representations to multiple learning targets. The secondary learning target of MTNL: rating regression of reviews, is considered a key component leading to higher performance, as it can help train the under-layer embeddings more sufficiently and prevent the entire model from over-fitting to the main target: helpfulness identification of reviews. To validate our hypothesis, we compare the performance of MTNL and STNL on the two datasets, i.e., Clothes and Electronics. Table IV illustrates that MTNL consistently surpasses STNL measured by all standard metrics in various underlying neural architectures, i.e. w/o char embeddings, w/o word embeddings, w/o attention mechanism, and etc. Moreover, the star rating of each review can also be regarded as a one-dimensional feature concatenated into the attention-based representation (i.e., $\mathbf{x}$ in Eq. (9) in STNL, instead of being used as the secondary learning target in MTNL. Thus, we conduct further experiments of STNL (with star ratings) on the two datasets, and Table IV shows that the performance of STNL (with star ratings) slightly improves compared with STNL, but still can not catch up with that of MTNL. The reason we suppose is that the star rating just provides only one among hundreds of features for the single-task classification but helps to generate hundreds of shared features as the secondary learning target.

D. Ability to Cross-domain Generalization

We build another two corpora with the Amazon Clothes and the Amazon Electronics datasets for cross-domain evaluation by simply exchanging their training sets. The two approaches, i.e., UGR + SVM and Fusion$_{all}$ + SVM, are chosen to compare with MTNL, as they show up the top-2 performance among all the mainstream methods on the task of identification of helpful reviews. Two groups of experiments are set by training the three models on different domains (different training sets), but testing their performance on the same one (the same test set). As shown in Table V, the F1 scores of the three models decrease on the cross-domain datasets. UGR + SVM and Fusion$_{all}$ + SVM drop significantly with 10.34% in F1 and 13.94% in F1. However, MTNL suffers little loss with 4.90% in F1 on average, and maintains comparable performance on the metrics of accuracy and AUC. Those experimental results demonstrate that MTNL has better generalization ability on cross-domain identification of helpful reviews.

VI. Conclusion and Future Work

This work contributes a novel architecture for automatically identifying helpful product reviews by means of a multi-task learning paradigm with deep neural networks. Benefited from the end-to-end learning fashion of our approach that automatically reconstructs effective features from the raw input of words and characters, it releases tedious labor of feature engineering with domain-specific knowledge as commonly adopted in the prior mainstream and state-of-the-art approaches. Moreover, the multi-task learning paradigm helps to produce more accurate predictions on helpful reviews by means of a secondary task of fitting their star ratings.

We construct two datasets containing reviews from different product categories in Amazon, where each review is collaboratively voted for helpfulness/unhelpfulness. Extensive experiments are conducted by comparing the performance of our method with other prior mainstream approaches, not only on specific-domain but also on cross-domain datasets. Results show that the MTNL architecture outperforms the prior state-of-the-art approach evaluated by several standard metrics.

In the future, we plan to explore more sophisticated designs of deep neural architecture, and to develop large-scale datasets for the task of helpful review identification.

ACKNOWLEDGMENT

This work was in part supported by the Joint Post-Doctoral Program of Baidu Inc. and Tsinghua University. The authors are grateful to the anonymous reviewers for their valuable and constructive comments.

REFERENCES

[1] J. McAuley and A. Yang, "Addressing complex and subjective product-related queries with customer reviews," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 625–635.

[2] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 507–517.

[3] S. M. Mudambi and D. Schuff, "What makes a helpful online review? a study of customer reviews on amazon.com," *MIS Quarterly*, vol. 34, no. 1, pp. 185–200, Mar. 2010.

[4] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, "Automatically assessing review helpfulness," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2006, pp. 423–430.

[5] W. Xiong and D. Litman, "Automatically predicting peer-review helpfulness," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2011, pp. 502–507.

[6] L. Martin and P. Pu, "Prediction of helpful reviews using emotions extraction," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 2014, pp. 1551–1557.

[7] Y. Yang, Y. Yan, M. Qiu, and F. Bao, "Semantic analysis and helpfulness prediction of text for online product reviews," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, 2015, pp. 38–44.

[8] H. Liu, Y. Gao, P. Lv, M. Li, S. Geng, M. Li, and H. Wang, "Using argument-based features to predict and analyse review helpfulness," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2017, pp. 1369–1374.

[9] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology*, vol. 2, no. 3, p. 27, 2011.

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[12] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[13] L. Deng, D. Yu *et al.*, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[14] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[15] L. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, 2001.

[16] Y. LeCun and Y. Bengio, "The handbook of brain theory and neural networks," pp. 255–258, 1998.

[17] D. Bonadiman, A. E. Uva, and A. Moschitti, "Multitask learning with deep neural networks for community question answering," *CoRR*, vol. abs/1702.03706, 2017.

[18] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang, "Representation learning using multi-task deep neural networks for semantic classification and information retrieval," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2015, pp. 912–921.

[19] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[20] M. McGlohon, N. Glance, and Z. Reiter, "Star quality: Aggregating reviews to rank products and merchants," in *Proceedings of Fourth International Conference on Weblogs and Social Media*, 2010.

[21] A. S. Tsang and G. Prendergast, "Is a star worth a thousand words? the interplay between product-review texts and rating valences," *European Journal of Marketing*, vol. 43, no. 11/12, pp. 1269–1280, 2009.

[22] V. Bewick, L. Cheek, and J. Ball, "Statistics review 13: receiver operating characteristic curves," *Critical care*, vol. 8, no. 6, p. 508, 2004.

[23] Y. Hochberg and Y. Benjamini, "More powerful procedures for multiple significance testing," *Statistics in medicine*, vol. 9, no. 7, pp. 811–818, 1990.

[24] R. Carver, "The case against statistical significance testing," *Harvard Educational Review*, vol. 48, no. 3, pp. 378–399, 1978.

[25] W. Xiong and D. J. Litman, "Empirical analysis of exploiting review helpfulness for extractive summarization of online reviews." in *Proceedings of the 25th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2014, pp. 1985–1995.

[26] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[27] K. R. Scherer, "What are emotions? and how can they be measured?" *Social science information*, vol. 44, no. 4, pp. 695–729, 2005.

[28] P. J. Stone, R. F. Bales, J. Z. Namenwirth, and D. M. Ogilvie, "The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information," *Behavioral Science*, vol. 7, no. 4, pp. 484–498, 1962.

[29] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[30] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and qsar modeling," *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[31] J. H. Friedman, T. J. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[32] P. Li, "Robust logitboost and adaptive base class (abc) logitboost," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2010, pp. 302–311.

[33] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," *arXiv preprint arXiv:1510.03820*, 2015.

[34] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[35] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 5998–6008.

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

[38] J. Hu and P. Li, "Collaborative filtering via additive ordinal regression," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 243–251.

[39] P. C. Woodland, S. E. Johnson, P. Jourlin, and K. S. Jones, "Effects of out of vocabulary words in spoken document retrieval (poster session)," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2000, pp. 372–374.

[40] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proceedings of the 27th international conference on machine learning*. Omnipress, 2010, pp. 111–118.

[41] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in *Proceedings of the 2nd International Conference on Neural Information Processing Systems*. MIT Press, 1989, pp. 396–404.

[42] M. A. Ranzato, Y.-L. Boureau, and Y. LeCun, "Sparse feature learning for deep belief networks," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*. Curran Associates Inc., 2007, pp. 1185–1192.