An Attention Model of Customer Expectation to Improve Review Helpfulness Prediction

Xianshan Qu(✉), Xiaopeng Li, Csilla Farkas, and John Rose

CSE Department, University of South Carolina, Columbia, SC 29201, USA
xqu@email.sc.edu, xl4@email.sc.edu {farkas,rose}@cse.sc.edu

Abstract. Many people browse reviews online before making purchasing decisions. It is essential to identify the subset of helpful reviews from the large number of reviews of varying quality. This paper aims to build a model to predict review helpfulness automatically. Our work is inspired by the observation that a customer's expectation of a review can be greatly affected by review sentiment and the degree to which the customer is aware of pertinent product information. Consequently, a customer may pay more attention to that specific content of a review which contributes more to its helpfulness from their perspective. To model such customer expectations and capture important information from a review text, we propose a novel neural network which leverages review sentiment and product information. Specifically, we encode the sentiment of a review through an attention module, to get sentiment-driven information from review text. We also introduce a product attention layer that fuses information from both the target product and related products, in order to capture the product related information from review text. Our experimental results show an AUC improvement of 5.4% and 1.5% over the previous state of the art model on Amazon and Yelp data sets, respectively.

Keywords: Review helpfulness · Customer expectation · Attention mechanism

1 Introduction

E-commerce has become an important part of our daily life. Increasingly, people choose to purchase products online. According to a recent study [9], most online shoppers browse reviews before making decisions. It is essential for users to be able to find reliable reviews of high quality. Therefore, an automatic helpfulness evaluation mechanism is in high demand to help user evaluate these reviews.

Previous work typically derived useful information from different sources, such as a review content [6,15,28], metadata [4,15,17], and context [14,20,25]. However, such features are extracted from each source independently, without considering interactions. In particular, previous approaches do not take into

© Springer Nature Switzerland AG 2020
J. M. Jose et al. (Eds.): ECIR 2020, LNCS 12035, pp. 836–851, 2020.
https://doi.org/10.1007/978-3-030-45439-5_55

account the customer review evaluating process. A customer's perception of helpful information of a review is affected by the sentiment of the review and what the customer already knows about the product. Before reading a review text for a product, the customer is very likely to be aware of background information such as star rating, product attributes, etc. When a customer reads a review with a lower star rating, he may hold a negative opinion towards the item at first and mainly look into those aspects of the review supporting the lower star rating. Although review sentiment has been previously explored [7,15,17], previous work has not used review sentiment to identify useful information from review text. Moreover, the customer likely has some preconceptions of the product features they are most interested in. With these expectations in mind, the customer pays special attention to those aspects of the review text that they find most salient. There have been earlier efforts [4,6,13] at capturing useful information from a review by considering product information. However, the unique aspects of each product (different levels of importance of attributes, evaluation standard, etc.) were not fully identified in those efforts.

In order to address the above issues, we propose a novel neural network architecture to introduce sentiment and product information when identifying helpful content from a review text. The main contributions are summarized as follows:

- To our knowledge, we are the first to propose that customers may have different expectations for reviews that express different sentiments. We design a sentiment attention layer to model sentiment-driven changes in user focus on a review.
- We propose a novel product attention layer. The purpose of this layer is to automatically identify the important product-related attributes from reviews. This layer fuses information not only from related products, but also from the specific product.
- We evaluate the performance of our model on two real-world data sets: the Amazon data set and the Yelp data set. We consider two application scenarios: cold start and warm start. In the cold start scenario, our proposed model demonstrates an AUC improvement of 5.4% and 1.5% on Amazon and Yelp data sets, respectively, when compared to the state of the art model. We also validate the effectiveness of each of the attention layers of our proposed model in cold start and warm start scenarios.

2 Related Work

Previous studies have concentrated on mining useful features from the content (i. e., the review itself) and/or the context (other sources such as reviewer or user information) of the reviews [6,10,13,15,18–20,25,27,28].

Content features have been extracted and widely utilized. They can be roughly broken down into the following categories: *structural features* [6,10,13,15,27,28], *lexical features* [10,15,27,28], *syntactic features* [10,15,27], *emotional features* [15,28], *semantic features* [6,10,13,15,18,27,28], and *argument*

features [12]. For instance, Kim et al. [10] investigated a variety of content features from Amazon product reviews, and found that features such as review length, unigrams and product ratings are most useful in measuring review helpfulness.

Context features have also been studied to improve helpfulness prediction [14, 20, 25]. For example, Lu et al. [14] examined social context that may reveal the quality of reviewers to enhance the prediction of the quality of reviews. While context information shows promise for improving helpfulness prediction, it may not be available across different platforms and is not appropriate for designing a universal model.

Deep Neural Networks have recently been proposed for helpfulness prediction of online reviews [1–4, 23]. Chen et al. [1] designed a word-level gating mechanism to represent the relative importance of each word. Fan et al. [3] proposed a multi-task paradigm to predict the star ratings of reviews and to identify the helpful reviews more accurately. They also utilized the metadata of the target product in addition to the textual content of a review to better represent a review [4].

The methods summarized above are representative of the research progress in review helpfulness prediction. Sentiment and product information have been explored previously [4, 7, 10, 15]. With respect to sentiment, Martin and Pu [15] extracted emotional words from review text to serve as important parameters for helpfulness prediction. However, previous research has not taken into account differences in customer expectations that can result from review sentiment perception. With respect to product information, Fan et al. [4] tried to better

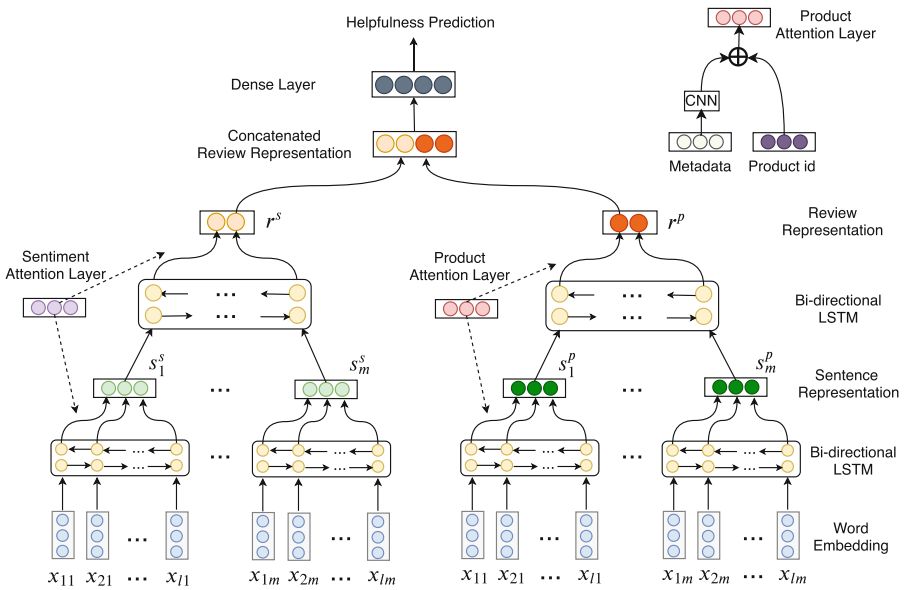

Fig. 1. The architecture of HSAPA.

represent the salient information in reviews by considering the metadata information (title, categories) of the target product. However, this information can be quite similar for products of the same type, so the unique aspects of each product (different degrees of importance of attributes, evaluation standard, etc.) can not be fully captured from reviews. Wu et al. [26] presented an architecture similar to the one we propose here. However, the design of the sentiment attention layer and product attention layer in our architecture is different from their attention layers. Moreover, their architecture is intended for classifying review sentiment.

3 Our Proposed Model

Our model, shown in Fig. 1, is built upon a hierarchical bi-directional LSTM. We incorporate sentiment and product information to improve review representations through two attention layers. As the main components of our model are the Hierarchical bi-directional LSTM, the Sentiment Attention layer, and the Product Attention layer, we refer to our model as HSAPA.

3.1 Hierarchical Bi-directional LSTM

A bi-directional LSTM model is able to learn past and future dependencies. This provides a better understanding of context [16]. The hierarchical architecture include two levels: the word level and the sentence level. These levels learn dependencies between words and sentences, respectively.

Word Encoder. A bi-directional LSTM consists of two LSTM networks that process data in opposite directions. At the word level, we feed the embedding of each word into a unit of both LSTMs, and get two hidden states. We then concatenate these two hidden states as a representation of a word. The process is defined as: $\overleftarrow{h}_{ij} = \overleftarrow{LSTM}(x_{ij})$, $\overrightarrow{h}_{ij} = \overrightarrow{LSTM}(x_{ij})$, $h_{ij} = [\overleftarrow{h}_{ij}, \overrightarrow{h}_{ij}]$, where x_{ij} is the embedding vector of the ith word of the jth sentence. $\overrightarrow{h}_{ij}$ and $\overleftarrow{h}_{ij}$ are hidden states learned from bi-directional LSTM. The state h_{ij} is the concatenation of these hidden states for the word x_{ij}.

Sentence Encoder. At the sentence level, a sentence representation is learned through an architecture similar to that used for the word level: $\overleftarrow{h}_j = \overleftarrow{LSTM}(s_j)$, $\overrightarrow{h}_j = \overrightarrow{LSTM}(s_j)$, $h_j = [\overleftarrow{h}_j, \overrightarrow{h}_j]$, where s_j refers to a weighted representation of the jth sentence after applying the attention layer. The state h_j is the final representation for the sentence s_j by concatenating the hidden states $\overrightarrow{h}_j$ and $\overleftarrow{h}_j$.

3.2 Sentiment Attention Layer

For reviews that express different types of sentiment (positive, negative, etc.), customers may have different expectations, and attend to different words or sentences of a review. Consider the following example:

> I loved the simplicity of the mouse, ... and it was very comfortable ... *About 4 months of owning the mouse the scroll wheel seemed to be in always clicked in position, and would only stop after clicking it down hard for a couple seconds ...*

The above review has a star rating of 2 out of 5. For a review with an overall negative sentiment like this, we may pay more attention to its descriptions of bad aspects of the product than we do to the good aspects. Therefore, each word/sentence may contribute unequally to the helpfulness of a review, with regard to its sentiment.

In order to learn the sentiment-influenced importance of each word/sentence, we propose a custom attention layer. In this layer, we use an embedded vector to represent each type of sentiment. We use the star rating (ranging from 1 to 5) of each review to indicate its sentiment, and map each discrete star rating into a real-valued and continuous vector *Sent*. This vector is initialized randomly, and updated gradually through the training process by reviews with the corresponding star rating. *Sent* can be interpreted as a high level representation of the sentiment-specific information. We measure the similarity between the sentiment and each word/sentence using a score function. The score function is defined as:

$$f(Sent, h_{ij}^s) = (v_w^s)^T \tanh(W_{wh}^s h_{ij}^s + W_{ws}^s Sent + b_w^s), \tag{1}$$

where v_w^s is a weight vector, and $(v_w^s)^T$ indicates its transpose, W_{wh}^s and W_{ws}^s are weight matrices, and b_w^s is the bias vector. At the word level, the input to the score function is the abstract sentiment representation *Sent* and the hidden state of the ith word in the jth sentence h_{ij}^s. Next, we use the softmax function to normalize the scores to get the attention weights:

$$\alpha_{ij}^s = \frac{\exp(f(Sent, h_{ij}^s))}{\sum_{k=1}^{l} \exp(f(Sent, h_{kj}^s))}, \tag{2}$$

α_{ij}^s is the attention weight for the word representation h_{ij}^s.

The sentence representation is a weighted aggregation of word representations, the jth sentence is represented as Eq. 3. The number of words in the jth sentence is denoted by l. The representation of a review is also a weighted combination of sentence representations defined as Eq. 4, where h_j^s is the hidden state of the jth sentence s_j^s, which is learned through the bi-directional LSTM. The value m refers the number of sentences in a review.

$$s_j^s = \sum_{i=1}^{l} \alpha_{ij}^s h_{ij}^s \ . \tag{3}$$

$$r^s = \sum_{j=1}^{m} \beta_j^s h_j^s \ . \tag{4}$$

The value β_j^s indicates the corresponding attention score for h_j^s. The weight score β_j^s is calculated based on the score function $f(.)$ defined as:

$$f(Sent, h_j^s) = (v_s^s)^T \tanh(W_{sh}^s h_j^s + W_{ss}^s Sent + b_s^s) \ , \tag{5}$$

$$\beta_j^s = \frac{\exp(f(Sent, h_j^s))}{\sum_{k=1}^{m} \exp(f(Sent, h_k^s))}. \tag{6}$$

3.3 Product Attention Layer

As shown in the top right corner of Fig. 1, the Product Attention Layer consists of two components: related product information and unique product information. Metadata information is embedded and fed into a CNN model [11] to capture the related product information, and the product identifier is encoded to represent the unique product information.

3.3.1 Related Product Information

When reading a review, customers may refer to different attributes depending on the product the review references. For example, for a review of a mouse, we may expect to see the comments related to attributes such as scroll wheel, hand feel, etc. Such attributes are considered helpful and garner more attention.

We take advantage of the metadata information (such as title, product description, product category, etc.) of each product to learn common attributes shared by related products. We use the pretrained GloVe embedding [21] to initialize each token in the metadata into a 100-dimensional embedding. We extract important attributes from the metadata through a CNN model [11], which is widely used for different NLP tasks [8,24,30,31] such as text understanding, document classification, etc.

The CNN model consists of a convolution layer, a max-pooling layer, and a fully connected layer. In the convolution layer, each filter is applied to a window of words to generate the feature map. For example, we apply a filter $w \in \mathbb{R}^{hk}$ to a window of words $x_{i:i+h-1}$. Here k indicates the dimension of the word vector, and $x_{i:i+h-1}$ refers to the concatenation of h words from x_i to x_{i+h-1}. The context feature c_{ih} is generated as:

$$c_{ih} = \text{ReLU}(wx_{i:i+h-1} + b) \ , \tag{7}$$

where b is the bias item. A feature map of the text is then generated through $c_h = [c_{1h}, c_{2h}, ..., c_{nh}]$, where $c_{1h}, c_{2h}, ...c_{nh}$ refer to context features extracted from different sliding windows of the text, and c_h indicates the concatenation of these features. The feature map c_h is then fed into a max-pooling layer, and the maximum value is extracted as $c = max\{c_h\}$ as the important information extracted by a particular filter. A number of filters are used, and the extracted features are concatenated and fed into a fully connected layer to generate a vector $Prod_1$. $Prod_1$ is a representation of the important related product attributes in the metadata.

3.3.2 Unique Product Information

Although reviews for the same type of product may share the same important attributes, the degree of importance of these attributes may vary from product to product. Consider pet food for example. Some pet food may be of good quality and fair price, but the flavor may not appeal to a picky eater. Conversely, price may be the most salient feature for some brands.

In order to represent the unique characteristics of each product, the unique product identifier for each product is mapped into a continuous vector $Prod_2$. At the outset, $Prod_2$ is randomly initialized. During the training process, this vector is only updated when reviews specific to the product are used for training. Thus $Prod_2$ can be interpreted as a high level representation of product-specific information. The final product representation $Prod$ is generated by combining the two vectors: $Prod_1$ and $Prod_2$ as:

$$Prod = \tanh(W_1 Prod_1 + W_2 Prod_2 + b^p), \tag{8}$$

where W_1 and W_2 are weight matrices for $Prod_1$ and $Prod_2$ respectively, and b^p is the bias vector. We calculate the product attention weights based on the score function $f(.)$, and the input to the score function is the product representation $Prod$ and hidden state of a word h^p_{ij}:

$$f(Prod, h^p_{ij}) = (v^p_w)^T \tanh(W^p_{wh} h^p_{ij} + W^p_{wp} Prod + b^p_w), \tag{9}$$

where $(v^p_w)^T$ denotes the transpose of weight vector v^p_w, W^p_{wh} and W^p_{wp} are weight matrices, and b^p_w is the bias vector. Then we apply softmax function to get a normalized attention score α^p_{ij}. At the word level, the sentence representation is defined in Eq. 10, where α^p_{ij} indicates the product attention score of the word representation h^p_{ij}. The representation of a review can be obtained formally through Eq. 11, where β^p_j indicates the attention weight for hidden state of the jth sentence h^p_j.

$$s^p_j = \sum_{i=1}^{l} \alpha^p_{ij} h^p_{ij}, \tag{10}$$

$$r^p = \sum_{j=1}^{m} \beta^p_j h^p_j, \tag{11}$$

After applying the sentiment attention layer and the product attention layer separately, we obtain two different review representations r^s and r^p. These two representations are concatenated as the final representation of a review $r = [r^s, r^p]$. Then, we apply a fully connected layer on top of r, to classify the helpfulness of a review.

3.4 Loss Function

To minimize the difference between the predicted helpfulness value and the actual helpfulness label, we utilize cross entropy loss as the objective function. It is a

commonly used loss function for binary classification, and is defined as:

$$Loss_{task} = -\sum_{i=1}^{N}(y_i log(p(y_i)) + (1 - y_i) log(1 - p(y_i))), \tag{12}$$

where y_i indicates the actual helpfulness label, $p(y_i)$ indicates the probability of helpfulness. N is the number of training observations. We present details on how these y_i are assigned in the following section.

4 Evaluation

In this section, we evaluate performance of our architecture in two application scenarios: cold start scenario and warm start scenario. Correspondingly, we split the data into training and test data differently for the two scenarios.

Data Sets. We evaluate our model on two publicly available data sets. One data set originates from Amazon reviews and was released by McAuley [5]. The other data set is from the Yelp Dataset Challenge 2018 [29]. We pre-process the data the same way as Fan et al. [4]: First, we join the product review with corresponding metadata information. Second, we filter out the reviews that have no votes. Last, we label reviews that receive more than 75% helpful votes out of total votes as helpful, and label the remaining reviews as unhelpful.

Table 1. Statistics of **Amazon** data set.

Category	# Products	# training data	# test data
Books	1,442,166	8,821,657	2,202,121
Clothing ...	477,958	1,478,488	372,662
Electronics	278,556	2,575,592	642,424
Grocery & ...	96,320	437,253	109,019
Health & ...	147,610	1,082,862	277,408
Home & Kitchen	220,687	1,480,520	360,402
Movies & TV	157,669	2,031,602	525,598
Pet Supplies	56,157	360,381	88,094
Tools & ...	134,446	637,594	162,161
Total	3,011,569	18,905,949	4,769,889

Table 2. Statistics of **Yelp** data set.

Category	# Products	# training data	# test data
Beauty & Spas	17,298	162,111	41,458
Health & Medical	15,458	102,592	25,687
Home Services	16,888	116,310	27,222
Restaurants	55,332	1,479,587	346,440
Shopping	29,489	220,431	55,743
Total	134,465	2,081,031	496,550

Evaluation Metric. In this study we use the Receiver Operating Characteristic Area Under the Curve (ROC AUC) statistic to evaluate the performance of our proposed model. This is a standard statistic used in the machine learning community to compare models. It is a robust statistic where imbalanced data sets are involved, and is a good metric for our problem where there are nearly four times as many helpful reviews as unhelpful reviews.

4.1 Application Scenario 1: Cold Start

In practice, a new product may have not yet received any helpful votes. Therefore assessment standards can't be captured from past voting information and can lead to the cold start problem. To evaluate model performance in this scenario,

we randomly select 80% of the *products* and their corresponding reviews as the training data set. The remaining products and their reviews are employed as the test data set. Therefore, all of the reviews for a given product appear only in the training data set or test data set. Consequently, all products in this test data set face the cold start problem. The statistics of the two data sets are summarized in Tables 1 and 2. Even though the partitioning approach is the same as that reported by Fan et al. [4], a consequence of the random selection of products into test and training data sets is that the actual number of reviews differs from that of Fan et al. [4]. However, the difference is less than 1%, which is not statistically significant.

4.1.1 Model Comparison

We compare our proposed model with several baseline models. Below is a list of the hand-crafted features.

- Structural features (STR) as introduced by Xiong et al. [27] and Yang et al. [28], include the number of tokens, number of question sentences, the star rating. They are used to reveal a user's attitude towards a product.
- Emotional features (GALC) as introduced by Martin et al. [15] use the Geneva Affect Label Coder dictionary to define 36 emotion states.
- Lexical features (LEX) including unigrams and bigrams are commonly used and weighted by tf-idf to represent a text. They are usually used as baseline models [1,28].
- Semantic features (INQUIRER) employed by Yang et al. [28] leverage the existing linguistic dictionary INQUIRER to represent a review in semantic dimensions.

The baseline models that we use to compare our model are:

- Fusion (SVM) uses a Support Vector Machine to fuse features from the preceding feature list.
- Fusion (R.F.) uses a Random Forest to fuse features from the preceding feature list.
- Embedding-Gated CNN (EG-CNN) [1] introduces a word-level gating mechanism that weights word embeddings to represent the relative importance of each word.
- Multi-task Neural Learning (MTNL) [3] is based on a multi-task neural learning architecture with a secondary task that tries to predict the star ratings of reviews.
- Product-aware Review Helpfulness Net (PRH-Net) [4] is a neural network-based model that introduces target product information to enhance the representation of a review. Fan et al. evaluate this model on the two data sets we are using and claim that PRH-Net is the state of the art.

Table 3. Review helpfulness prediction of **Amazon** data set. The best performances are in bold.

Category (AC)	LEX	INQUIRER	FUSION (SVM)	FUSION (R.F.)	EG-CNN	MTNL	PRH-Net	HSAPA
AC1: Books	0.572	0.620	0.594	0.601	0.625	0.629	0.652	**0.712**
AC2: Clothing, Shoes & Jewelry	0.538	0.608	0.587	0.557	0.590	0.592	0.614	**0.679**
AC3: Electronics	0.555	0.627	0.584	0.588	0.615	0.618	0.644	**0.723**
AC4: Grocery & Gourmet Food	0.526	0.618	0.537	0.556	0.613	0.638	0.715	**0.718**
AC5: Health & Personal Care	0.533	0.617	0.599	0.565	0.617	0.624	0.672	**0.723**
AC6: Home & Kitchen	0.545	0.609	0.579	0.573	0.605	0.611	0.630	**0.697**
AC7: Movies & TV	0.562	0.637	0.605	0.617	0.648	0.652	0.675	**0.753**
AC8: Pet Supplies	0.542	0.603	0.548	0.558	0.580	0.619	0.679	**0.701**
AC9: Tools & Home Improv.	0.548	0.592	0.565	0.586	0.607	0.621	0.644	**0.699**
Average	0.547	0.615	0.578	0.578	0.611	0.623	0.658	**0.712**

Table 4. Review helpfulness prediction of **Yelp** data set. The best performances are in bold.

Category (YC)	LEX	INQUIRER	FUSION (SVM)	FUSION (R.F.)	EG-CNN	MTNL	PRH-Net	HSAPA
YC1: Beauty & Spas	0.500	0.570	0.521	0.541	0.571	0.581	0.642	**0.669**
YC2: Health & Medical	0.517	0.584	0.535	0.538	0.580	0.603	0.665	**0.683**
YC3: Home Services	0.528	0.627	0.584	0.588	0.563	0.618	0.732	**0.736**
YC4: Restaurants	0.516	0.582	0.569	0.554	0.581	0.605	0.658	**0.664**
YC5: Shopping	0.518	0.609	0.542	0.555	0.572	0.619	0.674	**0.695**
Average	0.516	0.584	0.541	0.544	0.573	0.601	0.674	**0.689**

4.1.2 Results and Findings

We use the same data sets as Fan et al. [4] in the cold start scenario. This allows us to directly compare the performance of our model with the results reported by Fan et al. [4]. We also randomly select 10% of the products and their corresponding reviews from the training set as a validation data set. We then performed a grid search of hyper-parameter space on the validation data set to determine the best choice of hyper-parameters. The models were then trained based on the entire training data set with these fixed hyper-parameters.

Tables 3 and 4 show the results on the Amazon data set and Yelp data set, respectively. In Table 3 we see that our model outperforms previous models on all categories of the Amazon data set. The average improvement in AUC is 5.4% over the next best model. We observe that the degree of improvement varies from category to category. In the categories AC3 (Electronics), our model achieves improvement of 7.9%. In contrast, for the category AC4 (Grocery & Gourmet Food), the improvement is only 0.3%. We note that the category AC4, has less data than most of other categories (Table 1). Only the category AC8 (Pet Supplies) contains fewer products and reviews. However, there are proportionally more reviews per product for the category AC8 than for the category AC4. We suspect that sentiment embedding and product embedding may not be learned

well with such limited and divergent data. Therefore the improvement is not as high as that for the other categories. The results for the yelp data set are presented in Table 4. We find that our model also outperforms the previous models in all categories. The average improvement in AUC is 1.5% over the next best model. We note that the overall improvement is not as high as that demonstrated in the Amazon data set. This may be due to the relatively small number of products and reviews in the yelp data set. With the exception of the category YC4 (Restaurants), the other categories have fewer products and reviews than all of categories of Amazon data set.

The comparison results presented in Tables 3 and 4 show that our model outperforms the baseline models in the cold start scenario. In order to examine the significance of the improvement of our proposed model, we conducted a one-tailed t-test. As we are not certain as to whether the values reported in Fan et al. [4] refer to variance or standard deviation, we performed three tests: the one-sample t-test, and the two sample t-test where we evaluated both interpretations (variance and standard deviation) of the values reported in Fan et al. [4]. In all cases, the statistical results validate that our method is significantly better than the other baselines ($p < 0.001$).

In order to tease out the performance contribution of each of the components of our model, we evaluated different combinations of the components. The results are show in Table 5. Here HBiLSTM refers to the hierarchical bi-directional LSTM model without either of the attention layers. We use it as the baseline model for comparison. HSA refers to the combination of the HBiLSTM with the sentiment attention layer. HPA refers to the combination of the HBiLSTM with the product attention layer. Finally, HSAPA refers to the complete model which implements both attention layers. From Table 5, we see that adding a sentiment attention layer (HSA) to the base model (HBiLSTM) results in an average improvement in the AUC score of 2.0% and 2.6%, respectively on the Amazon and Yelp data sets. By adding a product attention layer (HPA) to the base model (HBiLSTM), the improvement is 0.7% and 1.3% on the Amazon and Yelp data sets respectively. Combining all three components results in an even larger increase in AUC score, 3.4% and 4.8%, respectively on the Amazon and Yelp data sets. We observe a synergistic effect resulting from the addition of the two attention layers. We also note that in both data sets, the improvement from the product attention layer is lower than that from the sentiment attention layer. This may be due to the fact that in the cold start scenario we have no information about the target product. Possibly the helpful attributes shared by related products may not be sufficiently accurate.

In order to verify that the gain in AUC is a consequence of the additional attention layers and not simply a result of adding more parameters, we conducted additional experiments. We adjusted the hyper-parameters of the HBiLSTM, HSA and HPA models to ensure they have approximately the same number of parameters as the complete model HSAPA. For example, for the category Grocery in the Amazon data set, the number of parameters of the complete model HSAPA is 30,194,490. We increased the number of hidden units in the

Table 5. Performance of each model in the Cold Start Scenario.

Data set	HBiLSTM	HSA	HPA	HSAPA
Amazon	0.678	**0.698**	**0.685**	**0.712**
Yelp	0.641	**0.667**	**0.654**	**0.689**

Table 6. Performance of each model on **Yelp** data set in the Warm Start Scenario. YC1-YC5 are described in Table 4.

Category	HBiLSTM	HSA	HPA	HSAPA
YC1	0.626	0.642	0.627	0.694
YC2	0.638	0.651	0.701	0.728
YC3	0.642	0.666	0.718	0.742
YC4	0.622	0.645	0.632	0.666
YC5	0.680	0.696	0.722	0.728
AVG.	0.642	**0.660**	**0.680**	**0.712**

Table 7. Performance of each model on **Amazon** data set in the Warm Start Scenario. AC1-AC10 are described in Table 3.

Category	HBiLSTM	HSA	HPA	HSAPA
AC1	0.682	0.712	0.704	0.775
AC2	0.657	0.667	0.717	0.723
AC3	0.693	0.708	0.705	0.725
AC4	0.682	0.699	0.758	0.779
AC5	0.689	0.718	0.767	0.782
AC6	0.665	0.696	0.703	0.744
AC7	0.739	0.764	0.818	0.826
AC8	0.666	0.709	0.727	0.746
AC9	0.664	0.702	0.727	0.745
AVG.	0.681	**0.708**	**0.736**	**0.760**

other three models to create new models with approximately the same number of parameters HBiLSTM: 30,420,604, HSA: 30,424,204, HPA: 30,412,858. Recall that the selection of hyper-parameters was determined by using a grid-search of the hyper-parameter space. Not surprisingly, the new models with more parameters do not demonstrate an improvement in performance in comparison to the models with hyper-parameters determined by grid-search. Our proposed model demonstrates improved performance, not simply because of greater modelling power due to more parameters, but because of the leveraging of sentiment and product related information by the sentiment and product attention layers.

4.2 Application Scenario 2: Warm Start

The warm start scenario is another commonly seen scenario in which some reviews for products have user votes, while other reviews haven't yet received user votes. For this scenario, we randomly select 80% of the reviews as the training data, and use the remaining reviews as the test data. The data statistics are essentially the same in scenario 2 as that in scenario 1 (Tables 1 and 2). As 80% of the reviews for products are in training data set, this partitioning produces a warm start scenario.

We also evaluated the contribution of each attention layer in the warm start scenario. Tables 6 and 7 show that, on average, the addition of the sentiment layer (HSA) to the base model increases the AUC by 1.8% and 2.7% on Yelp and Amazon data sets, respectively. We also find that the addition of the product attention layer (HPA) to the base model increases the AUC by 3.8% and 5.5% on Yelp and Amazon data sets, respectively. Comparing the results from warm start scenario to cold start Scenario, we make the following observations. First, the average performance of the base model (HBiLSTM) is very similar in both scenarios. Second, the AUC improvement from the product attention layer (HPA) is higher than that from the sentiment attention layer (HSA) on

both data sets in the warm start scenario. This improvement is not seen in the case of the cold start scenario. In the cold start scenario the product embedding can only be learned from reviews of related products. In contrast, in the warm start scenario product information can be learned from both the target product and related products. This explains why the product attention layer can achieve better performance in warm start compared to cold start scenario. From the performance results described in the cold start scenario section (Tables 3 and 4), we see that HSAPA out-performs the other models. We also see that HSAPA has even better performance in the warm start scenario (Tables 6 and 7). In practice, one can expect a mix of cold and warm start scenarios where HSAPA can be expected to demonstrate superior performance than in cold start scenario.

Table 8. Highlighted words by sentiment and product attention scores in two review examples.

Ex 1 Product attention:

did not fit on any of the tub spouts and was unable to stretch it enough to work. Had to return

Ex 1 Sentiment attention (star rating: 1):

did not fit on any of the tub spouts and was unable to stretch it enough to work. Had to return

Ex 2 Product attention:

This is a great blade. Almost no sanding needed after use and they remain sharp after several uses . Don't use them on rough construction material if you want them to keep doing the job they were meant to do.

Ex 2 Sentiment attention (star rating: 5):

This is a great blade. Almost no sanding needed after use and they remain sharp after several uses . Don't use them on rough construction material if you want them to keep doing the job they were meant to do.

4.3 Visualization of Attention Layers

We demonstrate the visual examination of attention scores applied at the word level by randomly sampling two identical review examples (shown in Table 8). We use two colors: red and green to represent the sentiment attention scores and product attention scores respectively. The lightness/darkness of the color is proportional to the magnitude of the attention score. There are a few interesting patterns to note. First, for the sentiment attention layer, the words that are assigned large weights have sentiment that is close to the overall sentiment of the review. For instance, in the example 2 the overall sentiment of the review is positive (5 out of 5). Although there are several negative words such as "no" and "don't", the positive words/phrases like "great", "remain sharp" are still assigned higher attention weights. This observation is consistent with our previous hypothesis that the word importance in a review can be affected by review sentiment. Second, the attributes or descriptive words of an attribute of the product in a review text gain higher weights from the product attention layer. For instance, in the first example the descriptive words "fit", "enough" and the

noun "tub" are assigned relatively high attention scores. Third, the combination of the important words captured by two attention layers can give us a brief and thorough summary of a review. It may also visually explain why the combination of these two can achieve a better result compared to a single attention layer.

5 Conclusion

In this paper, we describe our analysis of review helpfulness prediction and propose a novel neural network model with attention modules to incorporate sentiment and product information. We also describe the results of our experiments in two application scenarios: cold start and warm start. In the cold start scenario, our results show that the proposed model outperforms PRH-Net, the previous state of the art model. The increase in performance, measured by AUC, as compared with PRH-Net is 5.4% and 1.5% on Amazon and Yelp data sets, respectively. Furthermore, we evaluate the effect of each attention layer of proposed model in both scenarios. Both attention layers contribute to the improvement in performance. In the warm start scenario, the product attention layer is able to attain better performance than in cold scenario since it has access to reviews for targeted products. In this work, we evaluate review helpfulness from the perspective of review quality. For future work, we may rank the helpfulness of reviews by incorporating a user's own preferences [22] in order to make personalized recommendations.

Acknowledgments. This work was partially supported by 2018 IBM Faculty Award to the University of South Carolina.

References

1. Chen, C., et al.: Multi-domain gated CNN for review helpfulness prediction. In: Proceedings of the 2019 World Wide Web Conference (2019)
2. Chen, C., Yang, Y., Zhou, J., Li, X., Bao, F.S.: Cross-domain review helpfulness prediction based on convolutional neural networks with auxiliary domain discriminators. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Short Papers), vol. 2 (2018)
3. Fan, M., Feng, Y., Sun, M., Li, P., Wang, H., Wang, J.: Multi-task neural learning architecture for end-to-end identification of helpful reviews. In: 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (2018)
4. Fan, M., Feng, C., Guo, L., Sun, M., Li, P.: Product-aware helpfulness prediction of online reviews. In: Proceedings of the 2019 World Wide Web Conference (2019)
5. He, R., McAuley, J.: 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 (2016)
6. Hong, Y., Lu, J., Yao, J., Zhu, Q., Zhou, G.: What reviews are satisfactory: novel features for automatic helpfulness voting. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (2012)

7. Huang, A.H., Chen, K., Yen, D.C., Tran, T.P.: A study of factors that contribute to online review helpfulness. Comput. Hum. Behav. **48**(C), 17–27 (2015)
8. Johnson, R., Zhang, T.: Effective use of word order for text categorization with convolutional neural networks. In: NAACL HLT 2015, the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2015)
9. Kats, R.: Surprise! most consumers look at reviews before a purchase (2018). https://www.emarketer.com/content/surprise-most-consumers-look-at-reviews-before-a-purchase. Accessed 20 Aug 2019
10. Kim, S.M., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (2006)
11. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014)
12. Liu, H., et al.: Using argument-based features to predict and analyse review helpfulness. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (2017)
13. Liu, J., Cao, Y., Lin, C.Y., Huang, Y., Zhou, M.: Low-quality product review detection in opinion summarization. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL) (2007)
14. Lu, Y., Tsaparas, P., Ntoulas, A., Polanyi, L.: Exploiting social context for review quality prediction. In: Proceedings of the 19th International Conference on World Wide Web (2010)
15. Martin, L., Pu, P.: Prediction of helpful reviews using emotions extraction. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (2014)
16. Melamud, O., Goldberger, J., Dagan, I.: context2vec: learning generic context embedding with bidirectional LSTM. In: CoNLL (2016)
17. Mudambi, S.M., Schuff, D.: What makes a helpful online review? A study of customer reviews on amazon.com. MIS Q. **34**(1), 185–200 (2010)
18. Mukherjee, S., Popat, K., Weikum, G.: Exploring latent semantic factors to find useful product reviews. In: Proceedings of the 2017 SIAM International Conference on Data Mining (2017)
19. Ocampo Diaz, G., Ng, V.: Modeling and prediction of online product review helpfulness: a survey. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2018)
20. O'Mahony, M.P., Smyth, B.: Learning to recommend helpful hotel reviews. In: Proceedings of the Third ACM Conference on Recommender Systems (2009)
21. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP) (2014)
22. Qu, X., Li, L., Liu, X., Chen, R., Ge, Y., Choi, S.H.: A dynamic neural network model for CTR prediction in real-time bidding. In: 2019 IEEE International Conference on Big Data (Big Data) (2019)
23. Qu, X., Li, X., Rose, J.R.: Review helpfulness assessment based on convolutional neural network. arXiv abs/1808.09016 (2018). http://arxiv.org/abs/1808.09016
24. Severyn, A., Moschitti, A.: Twitter sentiment analysis with deep convolutional neural networks. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (2015)

25. Tang, J., Gao, H., Hu, X., Liu, H.: Context-aware review helpfulness rating prediction. In: Proceedings of the 7th ACM Conference on Recommender Systems (2013)
26. Wu, Z., Dai, X., Yin, C., Huang, S., Chen, J.: Improving review representations with user attention and product attention for sentiment classification. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (2018)
27. Xiong, W., Litman, D.: Automatically predicting peer-review helpfulness. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers, vol. 2 (2011)
28. Yang, Y., Yan, Y., Qiu, M., Bao, F.: 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) (2015)
29. Yelp: Yelp dataset challenge (2018). https://www.yelp.com/dataset/challenge. Accessed 14 Sept 2018
30. Zhang, X., LeCun, Y.: Text understanding from scratch. arXiv abs/1502.01710 (2015)
31. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems 28 (2015)