



Review helpfulness evaluation and recommendation based on an attention model of customer expectation

Xianshan Qu¹ · Xiaopeng Li¹ · Csilla Farkas¹ · John Rose¹

Received: 30 June 2020 / Accepted: 2 December 2020 / Published online: 2 January 2021
© The Author(s), under exclusive licence to Springer Nature B.V. part of Springer Nature 2021

Abstract

With the fast growth of e-commerce, more people choose to purchase products online and browse reviews before making decisions. It is essential to identify helpful reviews, given the typical large number of reviews and the various range of quality. In this paper, we aim 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 for the task of identifying whether a review is helpful or not 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. We further validate the effectiveness of each attention layer of our model in two application scenarios. The results demonstrate that both attention layers contribute to the model performance, and the combination of them has a synergistic effect. We also evaluate our model performance as a recommender system using three commonly used metrics: NDCG@10, Precision@10 and Recall@10. Our model outperforms PRH-Net, a state-of-the-art model, on all three of these metrics.

Keywords Review helpfulness · Customer expectation · Attention mechanism · Review recommendation

✉ Xianshan Qu
xqu@email.sc.edu

Extended author information available on the last page of the article

1 Introduction

E-commerce has become an important part of our daily life. More and more, people choose to purchase products online. According to recent studies (Fullerton 2017; Kats 2018), most online shoppers browse reviews before making decisions. It is essential for users to be able to find reliable reviews of high quality. To this end, several websites have implemented a voting mechanism that allows users to give feedback for online reviews. However, it is likely that users have yet to provide feedback on initial product reviews, and in the case of older products, recently posted reviews may not receive votes due to their low exposure. Therefore, an automatic helpfulness evaluation mechanism is in high demand to help users evaluate these reviews.

Previous works typically derived useful information from different sources, such as review content (Hong et al. 2012; Martin and Pu 2014; Yang et al. 2015), metadata (Fan et al. 2019; Martin and Pu 2014; Mudambi and Schuff 2010), and context (Lu et al. 2010; O'Mahony and Smyth 2009; Tang et al. 2013). However, such features were extracted from each source independently, without considering possible interactions. In particular, previous approaches do not take into 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, the customer's expectations may be affected by the sentiment of the review. If the review gives a low star rating, he may hold a negative opinion towards the item at first and mainly look into those aspects of the review supporting the low star rating. 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. I'm very disappointed with the quality of the mouse. ...

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 (text in italics) 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. Although review sentiment has been previously explored (Huang et al. 2015; Martin and Pu 2014; Mudambi and Schuff 2010), previous works have not used review sentiment to identify useful information from review text.

In addition, 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. For example, for a review of a computer mouse, we may expect to see the comments related to attributes such as hand feel, ease of use, scroll wheel and so on. Such attributes are considered helpful and garner more attention. Moreover, although the attributes that customers are interested in may be quite similar, the degree of importance of these attributes may vary from product to product. Consider the above review for example, here scroll wheel may be the most salient feature for the mouse. There have been earlier efforts (Fan et al. 2019; Hong et al. 2012; Liu et al. 2007) 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.

Our research focus is on evaluating the helpfulness of online reviews. We have explored two tasks: first, given a review, we evaluate if it is helpful or not. Second, for each product, we recommend the top n helpful reviews for users. As described above, we have insights to improve the performance of review helpfulness evaluation from two perspectives. Therefore, we have to address the following two research questions: (a) Can the sentiment of a review be used to identify the helpful information from a review and improve review helpfulness evaluation? (b) Can product related attributes, especially the unique attributes of each product, be used to identify helpful information from a review and improve review helpfulness evaluation?

In this paper, we explore these research questions and address design and performance issues in previous approaches to evaluating the helpfulness of online reviews. We propose a novel neural network architecture to introduce sentiment and product information when identifying helpful content from a review text. First, we use a hierarchical bi-directional LSTM to generate sentence-level and review-level representations. Then we augment the model with two attention layers to encode the sentiment and product information, respectively, into the review representation. The sentiment attention layer captures the sentiment-influenced importance of each word/sentence in the review. The product attention layer is designed to capture important attributes of a review from both related products and the particular product under consideration. We combine the review representations learned from the two attention layers with the expectation that these representations will behave synergistically. This study extends the work in Qu et al. (2020), 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 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 both two scenarios.
- In addition, we evaluate the performance of our model from the perspective of recommendations based on three metrics: NDCG@10, Precision@10 and Recall@10. Our model outperforms the state-of-the-art model PRH-Net designed by Fan et al. (2019) on all three of these metrics.

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 (Hong et al. 2012; Kim et al. 2006; Liu et al. 2007; Martin and Pu 2014;

Mukherjee et al. 2017; Ocampo Diaz and Ng 2018; O’Mahony and Smyth 2009; Tang et al. 2013; Xiong and Litman 2011; Yang et al. 2015).

Content features have been extracted and widely utilized. They can be roughly broken down into the following categories: *structural features*, *lexical features*, *syntactic features*, *emotional features*, *semantic features*, and *argument features* (Hong et al. 2012; Kim et al. 2006; Liu et al. 2017, 2007; Martin and Pu 2014; Mukherjee et al. 2017; Xiong and Litman 2011; Yang et al. 2015). Structural features include the number of tokens and sentences, the percentage of question sentences, the star rating, and so on. They are related to structural properties and are used to reveal a user’s attitude towards a product. Lexical features including unigrams and bigrams are weighted by tf-idf to represent a text. Syntactic features, such as the number/percentage of the verbs and nouns in a review, are used to capture the linguistic properties. Emotional features usually adopt 20 emotion categories from the Geneva Affect Label Coder dictionary. The frequency of each emotion category and the number of non-emotional words are counted as emotional features. For semantic features, researchers leveraged the existing linguistic dictionary INQUIRER to represent a review in semantic dimensions (Yang et al. 2015). For argument features, researchers focused on the argumentative sentences in a review and examined them from different perspectives like component, token, letter, and position (Liu et al. 2017).

Prior works have generally investigated one or more content features. For instance, Kim et al. (2006) 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. Yang et al. (2015) mainly exploited two semantic features (i. e., Linguistic Inquiry and Word Count, and General Inquirer) to analyze and predict helpful reviews. Martin and Pu (2014) proposed that emotional words play an important role in predicting helpfulness of review text. They extracted emotion from reviews by making use of GALC, a general lexicon of emotional words associated with a model representing 20 different categories, and results show that emotion based methods outperform previous structure based approach.

Context features have also been studied to improve helpfulness prediction (Lu et al. 2010; O’Mahony and Smyth 2009; Tang et al. 2013). For example, O’Mahony and Smyth (2009) combined features mined from the reviewer and the wider community reviewing activity, and features derived from the review text. Lu et al. (2010) examined social context that may reveal the quality of reviewers to enhance the prediction of the quality of reviews. Tang et al. (2013) identified the context information from the aspects of reviewers, raters and their relationship, and designed a context-aware model to predict review helpfulness. 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 (Chen et al. 2018, 2019; Fan et al. 2018, 2019; Qu et al. 2018). To tackle the problem of insufficient labeled data to build the review helpfulness model, Chen et al. (2018) proposed a model with a transfer learning module to adapt domain knowledge. The shared and domain-specific features are maintained separately by introducing adversarial and domain discrimination losses. Chen et al. (2019) designed a word-level gating mechanism to represent the relative importance of each word. Fan et al. (2018) 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 (Fan et al. 2019).

Available data sets Most prior research has utilized data sets constructed from Amazon product reviews (Chen et al. 2018, 2019; Hong et al. 2012; Kim et al. 2006; Liu et al. 2007; Mukherjee et al. 2017; Yang et al. 2015). The data set size varies from ~23K reviews of one product category (Liu et al. 2007) to ~2.9M reviews of five product categories (Chen et al. 2018). Some researchers also considered multiple data sources. Martin and Pu (2014) adopted three data sets collected from Amazon, Yelp, and TripAdvisor, respectively. But the data set from TripAdvisor is relatively small, containing only ~68K reviews. Tang et al. (2013) and Lu et al. (2010) constructed data sets from Ciao, a popular product review site. In contrast to Amazon, which allows users to give a binary vote for review helpfulness, Ciao supports scores ranging from 0 to 5 to indicate the helpfulness of a review. However, the Ciao data sets that they used are not publicly available. Fan et al. (2019) used two large-scale data sets: ~23.8M reviews from 9 Amazon product categories and ~2.6M reviews from 5 Yelp product categories. We employ the same data sets in our work as Fan et al. (2019).

The methods summarized above are representative of the research progress in review helpfulness prediction. Sentiment and product information have been explored previously (Fan et al. 2019; Huang et al. 2015; Kim et al. 2006; Martin and Pu 2014). With respect to sentiment, Martin and Pu (2014) extracted emotional words from review text to serve as important parameters for helpfulness prediction. Huang et al. (2015) found that the sentiment of a review is positively correlated with review helpfulness. 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. (2019) tried to better represent the salient information in reviews by considering the meta-data 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. (2018) proposed an architecture that is superficially similar to ours in the sense that both architectures are based on LSTM networks and attention layers. They utilized a user attention layer and a product attention layer to capture sentiment-related information. In contrast, we design a sentiment attention layer and a product attention layer to identify the helpful information from a review text. Consequently, the internal design of our attention layers are different from theirs as they serve completely different purposes.

3 Our proposed model

Our model is shown in Fig. 1. It is built upon a hierarchical bi-directional LSTM, which is a standard model for document understanding and classification (Liu and Guo 2019; Yang et al. 2016). We proposed two novel attention layers that incorporate sentiment and product information in order to improve review representations. The product attention layer is designed by fusing the information from both the target product and related products. The sentiment information is also encoded to capture helpful information from a review through the attention mechanism. After applying two attention layers separately, we get a combination of two review representations to predict the review helpfulness, which has a joint

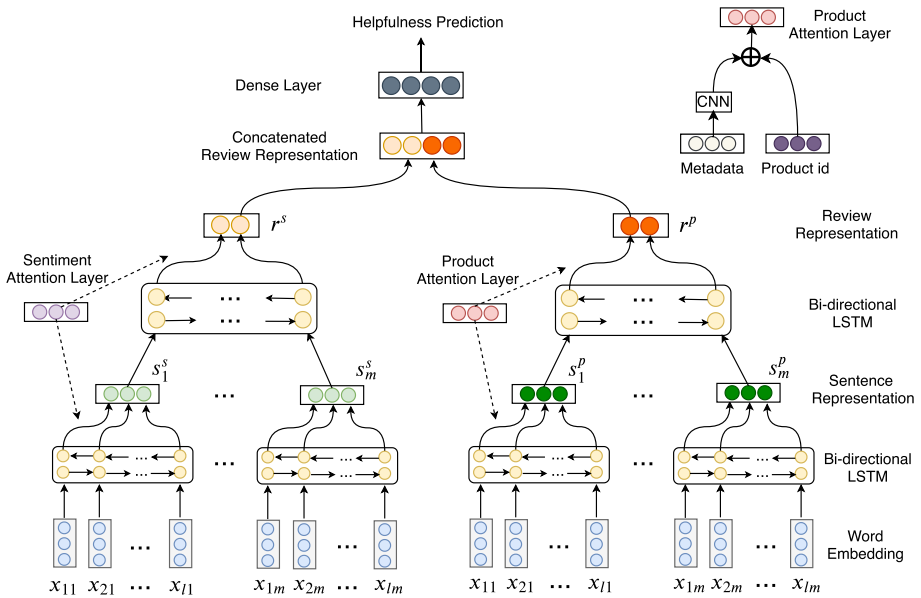


Fig. 1 The architecture of HSAPA

effect. 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

Our proposed model is based on a hierarchical bi-directional LSTM. A bi-directional LSTM model is able to learn past and future dependencies. This provides a better understanding of context (Melamud et al. 2016). The hierarchical architecture includes 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:

$$\vec{h}_{ij} = \overrightarrow{LSTM}(x_{ij}) \tag{1}$$

$$\overleftarrow{h}_{ij} = \overleftarrow{LSTM}(x_{ij}) \tag{2}$$

$$h_{ij} = [\vec{h}_{ij}, \overleftarrow{h}_{ij}] \tag{3}$$

where x_{ij} is the embedding vector of the i th word of the j th sentence. \vec{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:

$$\vec{h}_j = \overrightarrow{LSTM}(s_j) \tag{4}$$

$$\overleftarrow{h}_j = \overleftarrow{LSTM}(s_j) \tag{5}$$

$$h_j = [\vec{h}_j, \overleftarrow{h}_j] \tag{6}$$

where s_j refers to a weighted representation of the j th sentence after applying the attention layer. The state h_j is the the final representation for the sentence s_j by concatenating the hidden states \vec{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. In order to learn the sentiment-influenced importance of each word/sentence, we propose a custom attention layer.

In this attention layer, we use an embedded vector to represent each type of sentiment. We use the star rating of each review to indicate its sentiment, and map each discrete star rating into a real-valued and continuous vector *Sent*. For example, for Amazon reviews, a reader can give a star rating ranging from 1 to 5. In this case, we'll have 5 vectors that represent the five different types of sentiment. 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:

$$f(Sent, h_{ij}^s) = (v_w^s)^T \tanh(W_{wh}^s h_{ij}^s + W_{ws}^s Sent + b_w^s), \tag{7}$$

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 i th word in the j th sentence h_{ij}^s . Next, we use the softmax function to normalize the scores to get the attention weights:

$$\alpha_{ij}^s = \frac{\exp(f(Sent, h_{ij}^s))}{\sum_{k=1}^l \exp(f(Sent, h_{kj}^s))}, \tag{8}$$

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 j th sentence is represented as Eq. 9. The number of words in the j th sentence is denoted by

l . The representation of a review is also a weighted combination of sentence representations defined as Eq. 10, where h_j^s is the hidden state of the j th sentence s_j^s , which is learned through the bi-directional LSTM. The value m refers the number of sentences in a review.

$$s_j^s = \sum_{i=1}^l \alpha_{ij}^s h_{ij}^s. \quad (9)$$

$$r^s = \sum_{j=1}^m \beta_j^s h_j^s. \quad (10)$$

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 $f(\cdot)$ defined as:

$$f(\text{Sent}, h_j^s) = (v_s^s)^T \tanh(W_{sh}^s h_j^s + W_{ss}^s \text{Sent} + b_s^s), \quad (11)$$

$$\beta_j^s = \frac{\exp(f(\text{Sent}, h_j^s))}{\sum_{k=1}^m \exp(f(\text{Sent}, h_k^s))}. \quad (12)$$

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 (Kim 2014) 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 focus on different attributes depending on the product the review references. 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. Consider the following product description of a computer mouse:

Ergonomic shape - Ergonomically shaped design and soft rubber grips conform to your hand ...Interface - USB receiver...

Convenient controls - Easy-to-reach ...

Micro-precise scroll wheel - With more grooves per millimeter...

Long battery life - 3 year battery life ...

From this description, we want to learn common product related attributes such as “shape”, “interface”, “battery life”, “scroll wheel” etc. If these attributes appear in a review text, they may attract more customer attention.

In order to capture key information from the metadata, we make use of a CNN model (Kim 2014). This CNN model generalizes well for multiple NLP tasks such as text understanding (Zhang and LeCun 2015), document classification (Johnson and Zhang 2015; Severyn and Moschitti 2015; Zhang et al. 2015), etc. The CNN model can acquire important information

from a text. Moreover, it has a relatively simple architecture and fewer parameters compared to other models such as LSTM, Bi-LSTM, etc., and requires less training time.

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:

$$c_{ih} = \text{ReLU}(wx_{i:i+h-1} + b), \tag{13}$$

where b is the bias item.

We evaluated different approaches to initializing the word vector such as the pretrained Word2Vec embedding (Mikolov et al. 2013), the pretrained GloVe embedding (Pennington et al. 2014) and random initialized embedding, as well as different vector dimensions. The pretrained GloVe with 100 dimensions was able to achieve the best performance and required relatively less training time.

A feature map of the text is then generated through $c_h = [c_{1h}, c_{2h}, \dots, c_{nh}]$, where $c_{1h}, c_{2h}, \dots, c_{nh}$ 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. In order to represent the unique characteristics of each product, the unique product identifier for each product is mapped into a 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:

$$Prod = \tanh(W_1Prod_1 + W_2Prod_2 + b^p), \tag{14}$$

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 $f(\cdot)$, and the input to the score function is the product representation $Prod$ and hidden state of a word h_{ij}^p :

$$f(Prod, h_{ij}^p) = (v_w^p)^T \tanh(W_{wh}^p h_{ij}^p + W_{wp}^p Prod + b_w^p), \tag{15}$$

where $(v_w^p)^T$ denotes the transpose of weight vector v_w^p , W_{wh}^p and W_{wp}^p are weight matrices, and b_w^p is the bias vector. Then we apply softmax function to get a normalized attention score α_{ij}^p . At the word level, the sentence representation is defined in Eq. 16, where α_{ij}^p indicates the product attention score of the word representation h_{ij}^p . The representation of a review can be obtained formally through Eq. 17, where β_j^p indicates the attention weight for hidden state of the j th sentence h_j^p .

$$s_j^p = \sum_{i=1}^l \alpha_{ij}^p h_{ij}^p, \quad (16)$$

$$r^p = \sum_{j=1}^m \beta_j^p h_j^p, \quad (17)$$

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:

$$Loss_{task} = - \sum_{i=1}^N (y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))), \quad (18)$$

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 Experiment and results

This section focuses on evaluating our architecture with respect to review helpfulness. Given a review, we want to determine whether or not it is helpful. We first compared our model with competing models in prior works on two data sets. Then, we evaluated the performance of different components 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. Last, we compared the performance of our proposed model in both warm-start and cold-start scenarios.

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.

4.1 Data sets

We evaluate our model on two publicly available data sets. One data set originates from Amazon reviews and was released by Julian McAuley (He and McAuley 2016). The other data set is from the Yelp Dataset Challenge 2018 (Yelp 2018). We pre-process the data in the same way as Fan et al. (2019): 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.

We chose the same threshold of 75% as presented by Fan et al. (2019), in order to provide a fair comparison with their reported model performance. Moreover, the threshold of 75% makes more sense than a lower threshold such as 50%. Analysis of the data set shows that more than 80% of the reviews achieve a helpfulness vote ratio greater than 50%. In contrast, only around 60% of the reviews achieve a helpfulness vote ratio of more than 75%. If we chose a threshold of 50%, the problem would become much easier, as only the clearly unhelpful reviews would be labelled negative. More importantly, the majority of the reviews labelled as positive are not what we want. We want to identify only the most helpful reviews, in order to avoid having to read all of the reviews that would be labelled positive with a lower threshold.

4.1.1 Data set partition for cold start scenario

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 for each product category in both data sets. The remaining products and their reviews are employed as the test data set. The statistics of the two data sets are summarized in Tables 1 and 2. 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. Even though the partitioning approach is the same as that reported by Fan et al. (2019), 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. (2019). However, the difference is less than 1%, which is not statistically significant.

Table 1 Statistics of **Amazon** data set in **cold start** scenario

Category	Training Set			Test Set		
	# products	# samples	# positive samples	# products	# samples	# positive samples
Books	1,153,732	8,821,657	5,537,695	288,434	2,202,121	1,376,997
Clothing	382,366	1,478,488	1,076,066	95,592	372,662	271,737
Electronics	222,844	2,575,592	1,658,149	55,712	642,424	419,284
Grocery and Gourmet food	77,056	437,253	292,946	19,264	109,019	73,600
Health and Per- sonal Care	118,088	1,082,862	679,331	29,522	277,408	173,211
Home and Kitchen	176,549	1,480,520	1,091,194	44,138	360,402	266,020
Movies and TV	126,135	2,031,602	990,912	31,534	525,598	265,413
Pet Supplies	44,925	360,381	268,509	11,232	88,094	66,207
Tools and Home Improv.	107,556	637,594	450,303	26,890	162,161	114,439
Total	2,409,251	18,905,949	12,045,105	602,318	4,769,889	3,026,908

Table 2 Statistics of **Yelp** data set in **cold start** scenario

Category	Training Set			Test Set		
	# products	# samples	# positive samples	# products	# samples	# positive samples
Beauty and Spas	13,838	162,111	90,005	3,460	41,458	23,021
Health and Medical	12,366	102,592	66,753	3,092	25,687	16,899
Home Services	13,476	116,310	76,506	3,412	27,222	18,624
Restaurants	44,818	1,479,587	590,388	10,514	346,440	142,098
Shopping	23,591	220,431	101,967	5,898	55,743	27,262
Total	108,089	2,081,031	925,619	26,376	496,550	227,904

4.1.2 Data set partition for warm start scenario

The warm start scenario is another scenario in which some reviews for products have user votes, while other reviews haven't yet received user votes. In this scenario, we evaluated the different components in the proposed HSAPA model. Moreover, we compared the performance of HSAPA in the warm start scenario with that in the cold start scenario. We verified that our proposed model can achieve better performance in warm-start scenario where the unique product information can be captured.

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 shown in Tables 3 and 4. As 80% of the reviews for products are in training data set, this partitioning produces a warm start scenario. As in the cold-start scenario, we randomly selected 10% of the 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

Table 3 Statistics of **Amazon** data set in **warm start** scenario

Category	Training Set			Test Set		
	# products	# samples	# positive samples	# products	# samples	# positive samples
Books	1,326,937	8,819,022	5,531,753	695,623	2,204,756	1,382,939
Clothing	424,064	1,480,920	1,078,242	183,650	370,230	269,561
Electronics	257,439	2,574,412	1,661,946	143,507	643,604	415,487
Grocery and Gourmet food	87,851	437,017	293,236	43,055	109,255	73,310
Health and Personal Care	135,648	1,088,216	682,033	73,610	272,054	170,509
Home and Kitchen	201,516	1,472,737	1,085,771	103,922	368,185	271,443
Movies and TV	147,765	2,045,760	1,005,060	88,935	511,440	251,265
Pet Supplies	51,455	358,780	267,772	27,100	89,695	66,944
Tools and Home Improv.	122,065	639,804	451,793	60,408	159,951	112,949
Total	2,754,740	18,916,668	12,057,606	1,419,810	4,729,170	3,014,407

Table 4 Statistics of Yelp data set in warm start scenario

Category	Training Set			Test Set		
	# products	# samples	# positive samples	# products	# samples	# positive samples
Beauty and Spas	16,721	162,855	90,420	11,154	40,714	22,606
Health and Medical	14,873	102,623	66,921	9,315	25,656	16,731
Home Services	16,140	114,825	76,104	9,931	28,707	19,026
Restaurants	54,370	1,460,821	585,988	43,229	365,206	146,498
Shopping	28,493	220,939	103,383	17,887	55,235	25,846
Total	130,597	2,062,063	922,816	91,516	515,518	230,707

model with fixed hyper-parameters for each category are then trained on the entire training data set.

4.2 Model comparison

4.2.1 Competing models

We compare our proposed model with several baseline models in the cold start scenario. Two of the models, Fusion (GALC) and Fusion (R.F.), rely on hand-crafted features. The list of hand crafted features are Structural features (STR), Emotional features (GALC), Lexical features (LEX) and Semantic features (INQUIRER). These features were described earlier in Sect. 2. 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) (Chen et al. 2019) introduces a word-level gating mechanism that weights word embeddings to represent the relative importance of each word.
- Multi-task Neural Learning (MTNL) (Fan et al. 2018) 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) (Fan et al. 2019) 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.

The source code of the models listed above is not available. In their paper, Fan et al. (2019) implemented these models (Fusion SVM, Fusion R.F., EG-CNN, and MTNL) and reported a comparison of the results on two data sets with their own model (PRH-Net). These two data sets are publicly available (He and McAuley 2016; Yelp 2018). Therefore, we conducted experiments on the same data sets, and compare the performance of our model with the results reported in Fan et al. (2019)

4.2.2 Training settings

The training is based on the data sets in Tables 1 and 2. We use the same data sets and same partition approach as Fan et al. (2019). This allows us to directly compare the performance of our model with the results reported by Fan et al. (2019). We 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. These hyper-parameters include number of hidden units for each LSTM cell, embedding dimension for each word, learning rate, number of epochs and so on. These hyper-parameters were optimized on a per category basis. The models were then trained based on the entire training data set with these fixed hyper-parameters.

4.2.3 Results and findings

Tables 5 and 6 show the results on the Amazon data set and Yelp data set, respectively. In Table 5 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 and 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 6. 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 5 and 6 show that our model outperforms the baseline models in the cold start scenario.

4.2.4 Significance test

We further evaluate the significance of the improvement of our proposed model by conducting a one-tailed t test. We ran each model 20 times, and the number of degrees of freedom is 19.

We compared the result of our model (HSAPA) with the state of the art model (PRH-Net) in the code start scenario. The t test results for the Amazon and Yelp data sets are shown in Tables 7 and 8, respectively. In the second column of these tables, we show the average accuracy value and standard deviation for each category for our model. The last two columns show the calculated t value and the corresponding p value respectively. In the case of the Amazon data set (Table 7), the statistical results demonstrate that our method is significantly better than the state of the art model with a p value of 0.0005 for all categories

Table 5 Review helpfulness prediction of Amazon data set

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 and 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 and Gourmet Food	0.526	0.618	0.537	0.556	0.613	0.638	0.715	0.718
AC5: Health and Personal Care	0.533	0.617	0.599	0.565	0.617	0.624	0.672	0.723
AC6: Home and Kitchen	0.545	0.609	0.579	0.573	0.605	0.611	0.630	0.697
AC7: Movies and 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 and 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

The best performances are in bold

Table 6 Review helpfulness prediction of Yelp data set

Category (YC)	LEX	INQUIRER	FUSION (SVM)	FUSION (R.F.)	EG-CNN	MTNL	PRH-Net	HSAPA
YC1: Beauty and Spas	0.500	0.570	0.521	0.541	0.571	0.581	0.642	0.669
YC2: Health and 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

The best performances are in bold

Table 7 *t* test results for HSAPA and PRH-Net on Amazon data set

Category (AC)	PRH-Net	HSAPA	t-value	<i>p</i> value (<)
AC1: Books	0.652 ± 0.023	0.712 ± 0.044	6.098	0.0005
AC2: Clothing, Shoes and Jewelry	0.614 ± 0.006	0.679 ± 0.032	9.084	0.0005
AC3: Electronics	0.644 ± 0.017	0.723 ± 0.026	13.588	0.0005
AC4: Grocery and Gourmet Food	0.715 ± 0.077	0.718 ± 0.019	0.706	0.25
AC5: Health and Personal Care	0.672 ± 0.048	0.723 ± 0.047	4.853	0.0005
AC6: Home and Kitchen	0.630 ± 0.019	0.697 ± 0.029	10.332	0.0005
AC7: Movies and TV	0.675 ± 0.023	0.753 ± 0.051	6.840	0.0005
AC8: Pet Supplies	0.679 ± 0.060	0.701 ± 0.013	7.568	0.0005
AC9: Tools and Home Improv.	0.644 ± 0.023	0.699 ± 0.035	7.028	0.0005

Table 8 *t* test results for HSAPA and PRH-Net on Yelp data set

Category (YC)	PRH-Net	HSAPA	t-value	<i>p</i> value (<)
YC1: Beauty and Spas	0.642 ± 0.061	0.669 ± 0.018	6.708	0.0005
YC2: Health and Medical	0.665 ± 0.069	0.683 ± 0.022	3.659	0.001
YC3: Home Services	0.732 ± 0.129	0.736 ± 0.029	0.617	0.5
YC4: Restaurants	0.658 ± 0.053	0.664 ± 0.025	1.073	0.15
YC5: Shopping	0.674 ± 0.055	0.695 ± 0.036	2.609	0.01

Table 9 Performance of our models with different components

Data Set	Cold Start Scenario				Warm Start Scenario			
	HBiLSTM	HSA	HPA	HSAPA	HBiLSTM	HSA	HPA	HSAPA
Amazon	0.678	0.698	0.685	0.712	0.681	0.708	0.736	0.760
Yelp	0.641	0.667	0.654	0.689	0.642	0.660	0.680	0.712

The best performance for each scenario is indicated in bold

except for category AC4. In contrast to the other categories, category AC4 is a relatively small data set (437,253) representing a relatively large number of products (96,320). These results suggest that the model we propose performs better on larger data sets. Even in the case of category AC4, our model still achieves result comparable to the state of the art model. In the case of the Yelp data set (Table 8) the improvement in performance is not as uniformly statistically significant. This may be a consequence of the relatively small size of the training data set. Nonetheless, the results are still statistically significant with $p \leq 0.01$ for categories YC1, YC2, and YC5.

4.3 Evaluating different components of HSAPA

In order to tease out the performance contribution of each of the components of our model, we evaluated the different components of HSAPA model. We report the average

results across all categories on the Amazon and Yelp data sets for the models with different components in Table 9. 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. For this evaluation, we test the above models in both the cold start and warm start scenarios. The models are trained respectively on the data sets for both two scenarios, and the hyper-parameters are tuned to achieve the best AUC for each model.

In the cold start scenario, from Table 9, 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 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 the warm start scenario, we also evaluated the contribution of each attention layer and the combination of the two attention layers of the proposed HSAPA model. From Table 9, we see that 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. And 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.

In summary, we observe a synergistic effect resulting from the addition of the two attention layers, for both two scenarios. Comparing the two scenarios in Table 9, we have two additional observations. First, the average performance of the base model (HBiLSTM) is very similar in both scenarios. Second, adding the product attention layer (HPA) leads to higher improvements than adding the sentiment attention layer (HSA) on both data sets in the warm start scenario. But it's different for the case of cold

Table 10 Performance of HSAPA on **Amazon** data set in the cold start and warm start scenarios

Category	HSAPA (cold)	HSAPA (warm)
AC1: Books	0.712	0.775
AC2: Clothing, Shoes and Jewelry	0.679	0.723
AC3: Electronics	0.723	0.725
AC4: Grocery and Gourmet Food	0.718	0.779
AC5: Health and Personal Care	0.723	0.782
AC6: Home and Kitchen	0.697	0.744
AC7: Movies and TV	0.753	0.826
AC8: Pet Supplies	0.701	0.746
AC9: Tools and Home Improv.	0.699	0.745
Average	0.712	0.760

The best average performance over the entire Amazon dataset is indicated in bold

Table 11 Performance of HSAPA on **Yelp** data set in the cold start and warm start scenarios

Category	HSAPA (cold)	HSAPA (warm)
YC1: Beauty and Spas	0.669	0.694
YC2: Health and Medical	0.683	0.728
YC3: Home Services	0.736	0.742
YC4: Restaurants	0.664	0.666
YC5: Shopping	0.695	0.728
Average	0.689	0.712

The best average performance over the entire Yelp dataset is indicated in bold

start scenario. Here, separately adding the product attention layer (HPA) results in less of an improvement than does adding the sentiment attention layer (HSA).

4.4 Performance comparison of HSAPA in two scenarios

We further compared the performance of our proposed model HSAPA in two scenarios: warm start and cold start. Tables 10 and 11 show the results of HSAPA on each category of Amazon and Yelp data sets for the two scenarios. We see that HSAPA in the warm start scenario outperforms HSAPA in the cold start scenario on most categories of the two data sets. In the cold start scenario, the product embedding can only be learned from reviews of related products. In contrast, product information can be learned from both the target product and related products in the warm start scenario. This explains why HSAPA model can

Table 12 The *t* test results of HSAPA on **Amazon** data set for two scenarios

Category	HSAPA (cold)	HSAPA (warm)	t-value	<i>p</i> value (<)
AC1: Books	0.712 ± 0.044	0.775 ± 0.036	4.956	0.0005
AC2: Clothing, Shoes and Jewelry	0.679 ± 0.032	0.723 ± 0.029	4.556	0.0005
AC3: Electronics	0.723 ± 0.026	0.725 ± 0.033	0.213	0.5
AC4: Grocery and Gourmet Food	0.718 ± 0.019	0.779 ± 0.030	7.682	0.0005
AC5: Health and Personal Care	0.723 ± 0.047	0.782 ± 0.035	4.503	0.0005
AC6: Home and Kitchen	0.697 ± 0.029	0.744 ± 0.042	4.118	0.0005
AC7: Movies and TV	0.753 ± 0.051	0.826 ± 0.047	4.707	0.0005
AC8: Pet Supplies	0.701 ± 0.013	0.746 ± 0.019	8.742	0.0005
AC9: Tools and Home Improv.	0.699 ± 0.035	0.745 ± 0.028	4.590	0.0005

Table 13 *t* test results of HSAPA on **Yelp** data set for two scenarios

Category	HSAPA (cold)	HSAPA (warm)	t-value	<i>p</i> value (<)
YC1: Beauty and Spas	0.669 ± 0.018	0.694 ± 0.020	4.155	0.0005
YC2: Health and Medical	0.683 ± 0.022	0.728 ± 0.028	5.652	0.0005
YC3: Home Services	0.736 ± 0.029	0.742 ± 0.031	0.632	0.50
YC4: Restaurants	0.664 ± 0.025	0.666 ± 0.017	0.296	0.50
YC5: Shopping	0.695 ± 0.036	0.728 ± 0.025	3.367	0.005

achieve better performance in the warm start scenario. In practice, one can expect a mix of cold and warm start scenarios where HSAPA is expected to demonstrate superior performance than in cold start scenario.

We evaluated the significance of the performance difference of our proposed model in warm start and cold start scenarios by conducting a one-tailed t test. The t test results for the Amazon and Yelp data sets are shown in Tables 12 and 13, respectively. We find that the model in the warm start scenario can achieve significant improvement ($p < 0.005$) compared to that in the cold start scenario for most categories in both data sets. We also observe that, for Electronics in the Amazon data set, the performance is not significantly different. We suspect it is because the information captured from related products are enough to identify helpful information from a review text. For Home Services and Restaurants in the Yelp data set, we do not see a significant difference between the two scenarios. For these two categories, the significance result is consistent with the result shown in Table 8. It may be a consequence of the relatively small size of the training data set. In sum, the results for most categories demonstrate that the introduction of the product attention layer is able to capture unique product information in the warm start scenario and improve the accuracy of our model.

4.5 AUC gain

We observed that for both the cold-start and warm-start (see Table 9) scenarios, adding a sentiment attention layer (HSA) and a product attention layer (HPA) to the base model (HBiLSTM) results in improvement in the AUC score for both the Amazon and Yelp data sets. We want to verify that the gain in AUC is a consequence of the additional attention layers and not simply a result of adding more parameters for both scenarios. Therefore, we conducted an experiment to test the hypothesis that the observed improvement is due to the additional attention layers and not simply a result of adding more parameters.

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 in cold-start scenario is 30,194,490. We increased the number of hidden units in the 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.

5 Analysis

5.1 Visualization of attention layers

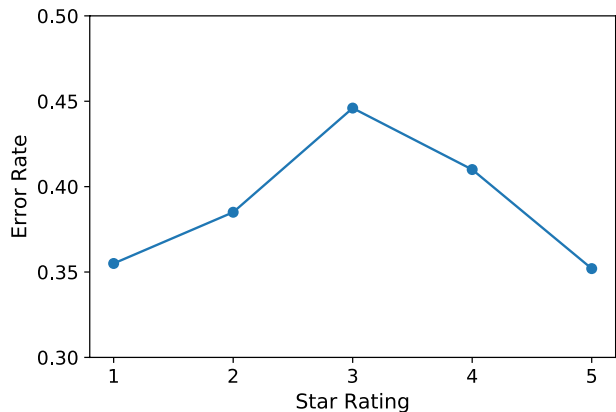
We demonstrate the visual examination of attention scores applied at the word level by randomly sampling three identical review examples (shown in Table 14). We use two

Table 14 Highlighted words by sentiment and product attention scores in three review examples

<p>did not fit on any of the tub spouts and was unable to stretch enough to work. Had to return</p> <p>Ex 1 Sentiment attention (star rating: 1):</p> <p>did not fit on any of the tub spouts and was unable to stretch it enough to work. Had to return</p> <p>Ex 2 Product attention:</p> <p>This is a great blade. Almost no sanding needed after use and the 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.</p> <p>Ex 2 Sentiment attention (star rating: 5):</p> <p>This is a great blade. Almost no sanding needed after use and the 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.</p> <p>Ex 3 Product attention:</p> <p>The knife itself seems well made and sturdy. Unfortunately I ignored all the feedback saying that the flashlight function would be DOA.</p> <p>Ex 3 Sentiment attention (star rating: 2):</p> <p>The knife itself seems well made and sturdy Unfortunately I ignored all the feedback saying that the flashlight function would be DOA.</p>

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. In the third example, the word “Unfortunately” is assigned more weight compared to the word “well”. 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

Fig. 2 The average error rate for each star rating on Amazon data set



review. It may also visually explain why the combination of these two can achieve a better result compared to a single attention layer.

5.2 Error analysis

In this section we present our analysis of misclassified reviews. First, from the perspective of sentiment, we calculated the average error rate for each star rating based on all categories of the Amazon data sets. The average error rate for each star rating is listed in Fig. 2. We found that for the star ratings of 3 and 4, we get relatively high error rates. We interpret that to mean that these star ratings reveal a neutral sentiment. The attention weights for words/sentences in these star ratings may not be that different. Consider the following example with a star rating of 3:

The product its self is great, it works wonderful, but getting to its original state is the tricky part not hard but time consuming!! It does its job and I got for a great price!

We analyzed the sentiment attention scores, and found that sentiment weights are almost equally distributed for each word in this review text such as the words 'great' and 'tricky'. Therefore, the effect of sentiment attention may not help a lot in this case.

We then calculated the average error rate for each product. We observed that products with more reviews seem to have smaller error rates. However, this finding is not consistent for all categories. It holds for some categories such as Home, Health and Tools, but fails for the Pet category. In the case of the Pet category, the error rate does not significantly decrease for products with large numbers of reviews.

5.3 Impact of sentiment on model performance

In this study, we make use of the star rating of each review to represent sentiment. In this section, we investigate the effect of different star rating combinations on model performance. We hypothesize that if the combinations of the star ratings represent very distinct categories, using sentiment derived from star ratings will help the model achieve better performance. The distributions of the different combinations of pairs of star ratings for the nine Amazon product categories are shown in Figs. 3, 4 and 5. These histograms depict the

Fig. 3 Histogram of different pairs of star rating combinations for categories: Home, Clothing, Grocery from the Amazon data set

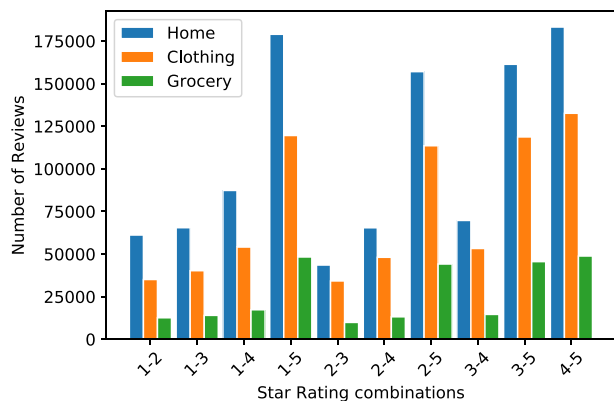


Fig. 4 Histogram of different pairs of star rating combinations for categories: Books, Electronics, Movies from the Amazon data set

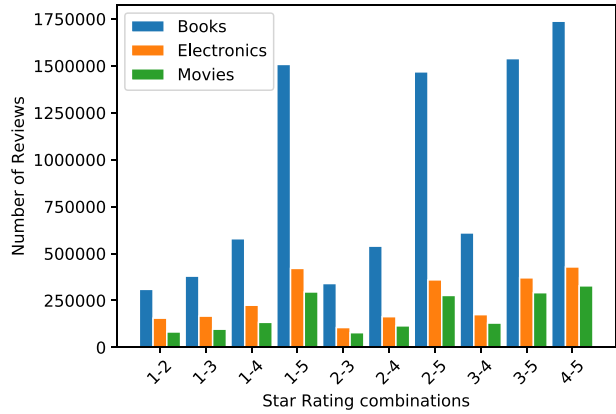
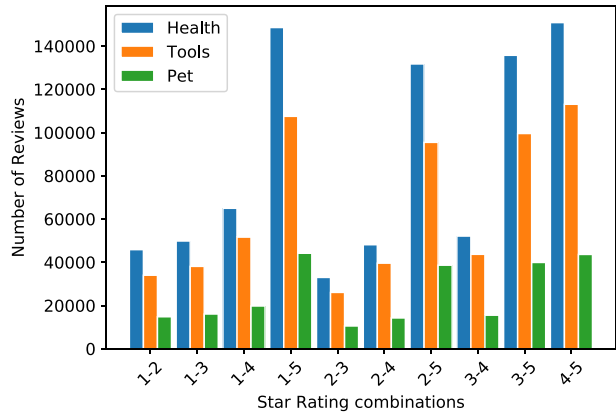


Fig. 5 Histogram of different pairs of star rating combinations for categories: Health, Tools, Pet from the Amazon data set



number of reviews for each of the possible pair combination (5 choose 2) of rating stars for each product category. From these figures, we find that there appear to be more reviews with the star rating of 5. Thus, there are more reviews for pair combinations of 5, i.e., 1–5, 2–5, 3–5, and 4–5. For other combinations, the number of reviews are similar, but noticeably fewer than combinations with 5.

We used HSAPA model trained in the warm start scenario to analyze the effect of sentiment on model performance. In the warm start scenario, we can avoid the possible effect of product attributes on model performance. The model is the same as the one we introduced in Sect. 4.4. (results shown in Table 10). The hyper-parameters (such as number of units for the Bi-LSTM module) are tuned to achieve the best AUC. This model was trained on reviews representing all star rating categories, i.e., one to five stars. We then analyzed the results by focusing on subsets of reviews corresponding to pairs of star ratings. For example, we considered the results for one star versus two stars, one star versus three stars, etc. We calculated the corresponding AUC value to examine the performance of different rating star combinations. If the reviews have two rating stars which are close, such as three and four, we consider this combination to have small variance in sentiment. In contrast, if the combination is a pair of rating stars of one and five, we consider this combination to have large variance in sentiment, i.e., very distinct sentiment categories.

Table 15 The AUC for Amazon reviews with different star rating distributions

Category (AC)	1–2	1–3	1–4	1–5	2–3	2–4	2–5	3–4	3–5	4–5	All
AC1: Books	0.705	0.721	0.766	0.760	0.691	0.739	0.728	0.712	0.720	0.687	0.775
AC2: Clothing, Shoes and Jewelry	0.682	0.699	0.725	0.715	0.677	0.701	0.703	0.692	0.698	0.672	0.723
AC3: Electronics	0.676	0.680	0.722	0.719	0.653	0.697	0.686	0.691	0.689	0.674	0.725
AC4: Grocery and Gourmet Food	0.703	0.714	0.777	0.765	0.682	0.732	0.719	0.714	0.719	0.675	0.779
AC5: Health and Personal Care	0.771	0.760	0.791	0.783	0.758	0.771	0.754	0.762	0.751	0.730	0.782
AC6: Home and Kitchen	0.690	0.699	0.744	0.746	0.694	0.728	0.717	0.719	0.712	0.686	0.744
AC7: Movies and TV	0.753	0.794	0.851	0.826	0.752	0.809	0.798	0.776	0.784	0.763	0.826
AC8: Pet Supplies	0.660	0.678	0.740	0.744	0.670	0.709	0.690	0.699	0.687	0.650	0.746
AC9: Tools and Home Improv.	0.693	0.707	0.746	0.746	0.703	0.729	0.722	0.725	0.720	0.703	0.745
Average	0.704	0.717	0.762	0.756	0.698	0.735	0.724	0.721	0.720	0.693	0.712

Table 15 shows the results of this experiment. The last column shows the AUC of reviews that have rating stars ranging from one to five, i.e., all rating stars are represented. The preceding columns present the AUC values produced on the predicted probabilities of reviews that contain only two rating star categories. For example, column “1–2” refers to the AUC for reviews with one or two rating stars. The result of this experiment shows that, on average, the combinations “1–4” and “1–5” exhibit the best performance. These two combinations represent distributions with large variance in sentiment. In contrast, the columns labeled “4–5”, “2–3”, and “1–2” show the AUC values resulting from combinations of pairs of rating stars that represent small variance in sentiment. The results validate our hypothesis that the rating stars of reviews affect model performance, and more divergent distributions of the rating stars can achieve better AUC values. Our model shows higher performance for reviews with very different rating stars. Intuitively, each product has an average rating, if a review has a sentiment that is inconsistent with the average rating, it may be considered as unhelpful. This finding is consistent with previous research (Hong et al. 2012) where the difference between current rating star and average rating was used as a feature. Those researchers found this feature improved performance.

6 Recommender system

The model we propose can also be utilized for recommendation purposes. For each product, in addition to evaluating the helpfulness of each review, we can also recommend the most helpful reviews for customers. We evaluate the performance of our model as a recommender system by comparing our results with those reported by Fan et al. (2019) for PRH-net.

6.1 Evaluation metrics

To evaluate the performance of our model for the recommendation problem, we use three commonly used metrics: $NDCG@n$, $Precision@n$ and $Recall@n$. Normalized Discounted Cumulative Gain (NDCG) is widely used to measure the quality and relevance of search algorithms in information retrieval. Here we apply it to evaluate the effectiveness of review ranking systems. It is computed as follows.

$$NDCG@n = \frac{DCG}{iDCG} = \frac{\sum_{i=1}^n \frac{2^{r(u_i)} - 1}{\log(1+i)}}{iDCG}, \quad (19)$$

where n is the number of reviews in the ranking list, i indicates the rank position of review u_i , $r(u_i) \in \{0, 1\}$ denotes the helpfulness of u_i (1: helpful, 0: unhelpful), and $iDCG$ is the DCG value computed based on the ideal ranking order of the same set of reviews. In our analysis, we chose n to be 10 for all three metrics.

6.2 Competing model

In Sect. 5, we show that our proposed HSAPA model outperforms all previous models at the task of identifying whether a review is helpful or not. Among all these models, PRH-Net achieves the best performance. Therefore, we compared the results of our model with that of the PRH-Net model on the task of recommending the top n reviews of each product.

Table 16 Comparison of HSAPA and PRH-Net models based on NDCG@10, Precision@10 and Recall@10 on each Amazon data category

Category (AC)	NDCG@10		Precision@10		Recall@10	
	PRH-Net	HSAPA	PRH-Net	HSAPA	PRH-Net	HSAPA
AC1: Books	0.531	0.608	0.741	0.932	0.548	0.694
AC2: Clothing, Shoes and Jewelry	0.637	0.652	0.775	0.874	0.603	0.559
AC3: Electronics	0.519	0.622	0.725	0.929	0.498	0.688
AC4: Grocery and Gourmet Food	0.580	0.693	0.760	0.882	0.590	0.552
AC5: Health and Personal Care	0.566	0.649	0.716	0.863	0.519	0.450
AC6: Home and Kitchen	0.702	0.763	0.785	0.899	0.530	0.560
AC7: Movies and TV	0.480	0.708	0.650	0.874	0.467	0.550
AC8: Pet Supplies	0.628	0.721	0.786	0.921	0.548	0.507
AC9: Tools and Home Improv.	0.592	0.752	0.772	0.919	0.571	0.661
Average	0.582	0.685	0.746	0.898	0.541	0.580

The best performances are in bold

We implemented the PRH-Net model based on the description in the paper Fan et al. (2019), and tuned the hyper-parameters to achieve the best AUC performance. Based on the predicted helpfulness for each review, we calculated the values of the three metrics.

6.3 Results

Table 16 lists the results of three metrics for each of the categories of the Amazon Data set for the HSAPA model and the PRH-Net model. From the table, we observe that on average, our model outperforms PRH-Net on each of the three metrics: NDCG@10, precision@10 and recall@10. Our proposed model achieves a precision of 89.8%. This means that in identifying the top 10 product reviews, our model is correct 89.8% of the time for this data set. The metric NDCG@10 gives us a more precise measure based on the position of the reviews we recommended. In terms of correctness of review position, our model also outperforms PRH-Net. When we look more closely at each category, we find that across all categories, our model achieves better NDCG@10 and precision@10 results. In contrast, the recall@10 results are mixed. The HSAPA model and the PRH-Net model achieve better performance on different categories. While it is usually desirable to strike a balance between precision and recall, it is often the case that it is not possible to attain equally high values for both. We find that our model achieves better precision@10 results than recall@10 results. The Precision@10 and NDCG@10 results indicate that most of the top 10 reviews for each product that our model recommends are of high quality and deemed to be helpful.

7 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. We also evaluate our model from the perspective of recommender systems with three commonly used metrics: NDCG@10, Precision@10 and Recall@10. Based on these results, our model outperforms the state-of-the-art model developed by Fan et al. (2019)

Our proposed HSAPA model is able to identify helpful information from a review text based on review rating star values and product metadata such as product descriptions. The HSAPA model not only identifies helpful reviews, but also recommends the top n helpful reviews for each product. This is quite useful when there are large numbers of reviews for a product. In this paper, we evaluate review helpfulness from the perspective of review quality. In the future, we may rank the helpfulness of reviews by incorporating a user's own preferences (Qu et al. 2019) in order to make personalized recommendations (Devlin et al. 2018).

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

References


- Chen, C., Qiu, M., Yang, Y., Zhou, J., Huang, J., Li, X., et al. (2019). Multi-domain gated CNN for review helpfulness prediction. In *Proceedings of the 2019 World Wide Web conference*.
- Chen, C., Yang, Y., Zhou, J., Li, X., & Bao, F. S. (2018). 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* (Vol. 2, Short Papers).
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. CoRR [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
- Fan, M., Feng, C., Guo, L., Sun, M., & Li, P. (2019). Product-aware helpfulness prediction of online reviews. In *Proceedings of the 2019 World Wide Web conference*.
- Fan, M., Feng, Y., Sun, M., Li, P., Wang, H., & Wang, J. (2018). 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)*.
- Fullerton, L. (2017). Online reviews impact purchasing decisions for over 93% of consumers, report suggests. Retrieved December 20, 2019, from <https://www.thedrum.com/news/2017/03/27/online-reviews-impact-purchasing-decisions-over-93-consumers-report-suggests>.
- He, R., & McAuley, J. (2016). 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*.
- Hong, Y., Lu, J., Yao, J., Zhu, Q., & Zhou, G. (2012). 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*.
- Huang, A. H., Chen, K., Yen, D. C., & Tran, T. P. (2015). A study of factors that contribute to online review helpfulness. *Computers in Human Behavior*, 48, 17–27.
- Johnson, R., & Zhang, T. (2015). 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*.
- Kats, R. (2018). Surprise! most consumers look at reviews before a purchase. Retrieved August 20, 2019, from <https://www.emarketer.com/content/surprise-most-consumers-look-at-reviews-before-a-purchase>.

- Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.
- Kim, S. M., Pantel, P., Chklovski, T., & Pennacchiotti, M. (2006). Automatically assessing review helpfulness. In *Proceedings of the 2006 conference on empirical methods in natural language processing*.
- Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. *Neurocomputing*, 337, 325–338.
- Liu, H., Gao, Y., Lv, P., Li, M., Geng, S., Li, M., & Wang, H. (2017). Using argument-based features to predict and analyse review helpfulness. In *Proceedings of the 2017 conference on empirical methods in natural language processing*.
- Liu, J., Cao, Y., Lin, C. Y., Huang, Y., & Zhou, M. (2007). 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)*.
- Lu, Y., Tsaparas, P., Ntoulas, A., & Polanyi, L. (2010). Exploiting social context for review quality prediction. In *Proceedings of the 19th international conference on World Wide Web*.
- Martin, L., & Pu, P. (2014). Prediction of helpful reviews using emotions extraction. In *Proceedings of the twenty-eighth AAAI conference on artificial intelligence*.
- Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: learning generic context embedding with bidirectional lstm. In *CoNLL*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 26, pp. 3111–3119). Red Hook: Curran Associates Inc.
- Mudambi, S. M., & Schuff, D. (2010). What makes a helpful online review? A study of customer reviews on amazon.com. *MIS Quarterly*, 34(1), 185–200.
- Mukherjee, S., Papat, K., & Weikum, G. (2017). Exploring latent semantic factors to find useful product reviews. In *Proceedings of the 2017 SIAM international conference on data mining*.
- Ocampo Diaz, G., & Ng, V. (2018). 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)*.
- O'Mahony, M. P., & Smyth, B. (2009). Learning to recommend helpful hotel reviews. In *Proceedings of the third ACM conference on recommender systems*.
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical methods in natural language processing (EMNLP)*.
- Qu, X., Li, L., Liu, X., Chen, R., Ge, Y., & Choi S. H. (2019). A dynamic neural network model for CTR prediction in real-time bidding. In *2019 IEEE international conference on big data (Big Data)*.
- Qu, X., Li, X., & Rose, J. R. (2018). Review helpfulness assessment based on convolutional neural network. [arXiv:1808.09016](https://arxiv.org/abs/1808.09016)
- Qu, X., Li, X., Farkas, C., & Rose, J. (2020). An attention model of customer expectation to improve review helpfulness prediction. In J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, & F. Martins (Eds.), *Advances in information retrieval* (pp. 836–851). Cham: Springer.
- Severyn, A., & Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*.
- Tang, J., Gao, H., Hu, X., & Liu, H. (2013). Context-aware review helpfulness rating prediction. In *Proceedings of the 7th ACM conference on recommender systems*.
- Wu, Z., Dai, X., Yin, C., Huang, S., & Chen, J. (2018). Improving review representations with user attention and product attention for sentiment classification. In *Proceedings of the thirty-second AAAI conference on artificial intelligence*.
- Xiong, W., & Litman, D. (2011). Automatically predicting peer-review helpfulness. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers—Volume 2*.
- Yang, Y., Yan, Y., Qiu, M., & Bao, F. (2015). 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)*.
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). *Hierarchical attention networks for document classification*. San Diego, CA: Association for Computational Linguistics.
- Yelp. (2018). Yelp dataset challenge. Retrieved September 19, 2018, from <https://www.yelp.com/dataset/challenge>.
- Zhang, X., & LeCun, Y. (2015). Text understanding from scratch. [arXiv:1502.01710](https://arxiv.org/abs/1502.01710).

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems* 28.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Xianshan Qu¹  · Xiaopeng Li¹ · Csilla Farkas¹ · John Rose¹

Xiaopeng Li
xl4@email.sc.edu

Csilla Farkas
farkas@cse.sc.edu

John Rose
rose@cse.sc.edu

¹ CSE Department, University of South Carolina, Columbia, SC 29201, USA