

# Rethinking Attribute Representation and Injection for Sentiment Classification

Reinald Kim Amplayo

Institute for Language, Cognition and Computation

School of Informatics, University of Edinburgh

reinald.kim@ed.ac.uk

## Abstract

Text attributes, such as user and product information in product reviews, have been used to improve the performance of sentiment classification models. The de facto standard method is to incorporate them as additional biases in the attention mechanism, and more performance gains are achieved by extending the model architecture. In this paper, we show that the above method is the least effective way to represent and inject attributes. To demonstrate this hypothesis, unlike previous models with complicated architectures, we limit our base model to a simple BiLSTM with attention classifier, and instead focus on *how* and *where* the attributes should be incorporated in the model. We propose to represent attributes as chunk-wise importance weight matrices and consider four locations in the model (i.e., embedding, encoding, attention, classifier) to inject attributes. Experiments show that our proposed method achieves significant improvements over the standard approach and that attention mechanism is the worst location to inject attributes, contradicting prior work. We also outperform the state-of-the-art despite our use of a simple base model. Finally, we show that these representations transfer well to other tasks<sup>1</sup>.

## 1 Introduction

The use of categorical attributes (e.g., user, topic, aspects) in the sentiment analysis community (Kim and Hovy, 2004; Pang and Lee, 2007; Liu, 2012) is widespread. Prior to the deep learning era, these information were used as effective categorical features (Li et al., 2011; Tan et al., 2011; Gao et al., 2013; Park et al., 2015) for the machine learning model. Recent work has used them to improve the overall performance (Chen et al.,

2016; Dong et al., 2017), interpretability (Amplayo et al., 2018a; Angelidis and Lapata, 2018), and personalization (Ficler and Goldberg, 2017) of neural network models in different tasks such as sentiment classification (Tang et al., 2015), review summarization (Yang et al., 2018a), and text generation (Dong et al., 2017).

In particular, user and product information have been widely incorporated in sentiment classification models, especially since they are important metadata attributes found in review websites. Tang et al. (2015) first showed significant accuracy increase of neural models when these information are used. Currently, the accepted standard method is to use them as additional biases when computing the weights  $a$  in the attention mechanism, as introduced by Chen et al. (2016) as:

$$a = \text{softmax}(v^\top e) \quad (1)$$

$$e = \tanh(Wh + W_u u + W_p p + b) \quad (2)$$

$$= \tanh(Wh + b_u + b_p + b) \quad (3)$$

$$= \tanh(Wh + b') \quad (4)$$

where  $u$  and  $p$  are the user and product embeddings, and  $h$  is a word encoding from BiLSTM. Since then, most of the subsequent work attempted to improve the model by extending the model architecture to be able to utilize external features (Zhu and Yang, 2017), handle cold-start entities (Amplayo et al., 2018a), and represent user and product separately (Ma et al., 2017).

Intuitively, however, this method is not the ideal method to represent and inject attributes because of two reasons. First, representing attributes as additional biases cannot model the relationship between the text and attributes. Rather, it only adds a user- and product-specific biases that are independent from the text when calculating the attention weights. Second, injecting the attributes in the attention mechanism means that user and prod-

<sup>1</sup> Model implementation and datasets are released here: <https://github.com/rktamplayo/CHIM>

uct information are only used to customize how the model choose which words to focus on, as also shown empirically in previous work (Chen et al., 2016; Ma et al., 2017). However, we argue that there are more intuitive locations to inject the attributes such as when contextualizing words to modify their sentiment intensity.

We propose to represent user and product information as weight matrices (i.e.,  $W$  in the equation above). Directly incorporating these attributes into  $W$  leads to large increase in parameters and subsequently makes the model difficult to optimize. To mitigate these problems, we introduce chunk-wise importance weight matrices, which (1) uses a weight matrix smaller than  $W$  by a chunk size factor, and (2) transforms these matrix into gates such that it corresponds to the relative importance of each neuron in  $W$ . We investigate the use of this method when injected to several locations in the base model: word embeddings, BiLSTM encoder, attention mechanism, and logistic classifier.

The results of our experiments can be summarized in three statements. First, our preliminary experiments show that doing bias-based attribute representation and attention-based injection is not an effective method to incorporate user and product information in sentiment classification models. Second, despite using only a simple BiLSTM with attention classifier, we significantly outperform previous state-of-the-art models that use more complicated architectures (e.g., models that use hierarchical models, external memory networks, etc.). Finally, we show that these attribute representations transfer well to other tasks such as product category classification and review headline generation.

## 2 How and Where to Inject Attributes?

In this section, we explore different ways on how to represent attributes and where in the model can we inject them.

### 2.1 The Base Model

The majority of this paper uses a base model that accepts a review  $\mathbf{x} = x_1, \dots, x_n$  as input and returns a sentiment  $y$  as output, which we extend to also accept the corresponding user  $u$  and product  $p$  attributes as additional inputs. Different from previous work where models use complex architectures such as hierarchical LSTMs (Chen et al., 2016; Zhu and Yang, 2017) and external memory

networks (Dou, 2017; Long et al., 2018), we aim to achieve improvements by only modifying how we represent and inject attributes. Thus, we use a simple classifier as our base model, which consists of four parts explained briefly as follows.

First, we embed  $\mathbf{x}$  using a word embedding matrix that returns word embeddings  $x'_1, \dots, x'_n$ . We subsequently apply a non-linear function to each word:

$$w_t = \tanh(W_{\text{emb}}x'_t + b_{\text{emb}}) \quad (5)$$

Second, we run a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) encoder to contextualize the words into  $h_t = [\vec{h}_t; \overleftarrow{h}_t]$  based on their forward and backward neighbors. The forward and backward LSTM look similar, thus for brevity we only show the forward LSTM below:

$$\begin{bmatrix} g_t \\ i_t \\ f_t \\ o_t \end{bmatrix} = \begin{bmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{bmatrix} W_{\text{enc}}[w_t; \vec{h}_{t-1}] + b_{\text{enc}} \quad (6)$$

$$c_t = f_t * c_{t-1} + i_t * g_t \quad (7)$$

$$\vec{h}_t = o_t * c_t \quad (8)$$

Third, we pool the encodings  $h_t$  into one document encoding  $d$  using attention mechanism (Bahdanau et al., 2015), where  $v$  is a latent representation of informativeness (Yang et al., 2016):

$$e_t = \tanh(W_{\text{att}}h_t + b_{\text{att}}) \quad (9)$$

$$a_t = \text{softmax}_t(v^\top e_t) \quad (10)$$

$$d = \sum_t (a_t * h_t) \quad (11)$$

Finally, we classify the document using a logistic classifier to get a predicted  $y'$ :

$$y' = \text{argmax}(W_{\text{cls}}d + b_{\text{cls}}) \quad (12)$$

Training is done normally by minimizing the cross entropy loss.

### 2.2 How: Attribute Representation

Note that at each part of the model, we see similar non-linear functions, all using the same form, i.e.  $g(f(x)) = g(Wx + b)$ , where  $f(x)$  is an affine transformation function of  $x$ ,  $g$  is a non-linear activation,  $W$  and  $b$  are weight matrix and bias parameters, respectively. Without extending the base model architecture, we can represent the attributes either as the weight matrix  $W$  or as the bias  $b$  to one of these functions by modifying them to accept  $u$  and  $p$  as inputs, i.e.  $f(x, u, p)$ .

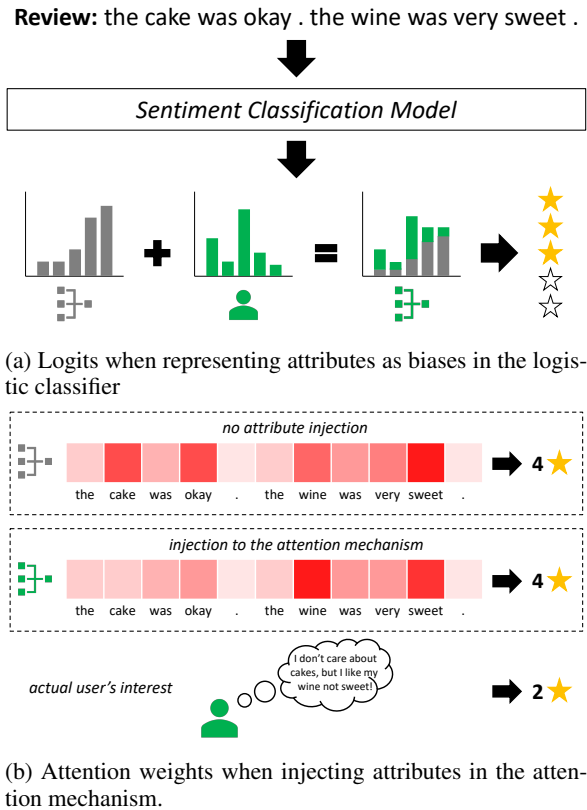


Figure 1: Illustrative examples of issues when representing attributes as biases and injecting them in the attention mechanism. The gray process icon indicates the model without incorporating attributes, while the same icon in green indicates the model customized for the green user.

**Bias-based** The current accepted standard approach to represent the attributes is through the bias parameter  $b$ . Most of the previous work (Chen et al., 2016; Zhu and Yang, 2017; Amplayo et al., 2018a; Wu et al., 2018) use Equation 2 in the attention mechanism, which basically updates the original bias  $b$  to  $b' = W_u u + W_p p + b$ . However, we argue that this is not the ideal way to incorporate attributes since it means we only add a user- and product-specific bias towards the goal of the function, *without* looking at the text. Figure 1a shows an intuitive example: When we represent user  $u$  as a bias in the logistic classifier, in which it means that  $u$  has a biased logits vector  $b_u$  of classifying the text as a certain sentiment (e.g.,  $u$  tends to classify texts as three-star positive), shifting the final probability distribution regardless of what the text content may have been.

**Matrix-based** A more intuitive way of representing attributes is through the weight matrix  $W$ . Specifically, given the attribute embeddings  $u$  and

$p$ , we linearly transform their concatenation into a vector  $w'$  of size  $D_1 * D_2$  where  $D_1$  and  $D_2$  are the dimensions of  $W$ . We then reshape  $w'$  into  $W'$  to get the same shape as  $W$  and replace  $W$  with  $W'$ :

$$w' = W_c[u; p] + b_c \quad (13)$$

$$W' = \text{reshape}(w', (D_1 \times D_2)) \quad (14)$$

$$f(x, u, p) = W'x + b \quad (15)$$

Theoretically, this should perform better than bias-based representations since direct relationship between text and attributes are modeled. For example, following the example above,  $W'x$  is a user-biased logits vector based on the document encoding  $d$  (e.g.,  $u$  tends to classify texts as two-star positive when the text mentions that the dessert was sweet).

However, the model is burdened by a large number of parameters; matrix-based attribute representation increases the number of parameters by  $|U| * |P| * D_1 * D_2$ , where  $|U|$  and  $|P|$  correspond to the number of users and products, respectively. This subsequently makes the weights difficult to optimize during training. Thus, directly incorporating attributes into the weight matrix may cause harm in the performance of the model.

**CHIM-based** We introduce **Chunk-wise Importance Matrix (CHIM)** based representation, which improves over the matrix-based approach by mitigating the optimization problems mentioned above, using the following two tricks. First, instead of using a big weight matrix  $W'$  of shape  $(D_1, D_2)$ , we use a chunked weight matrix  $C$  of shape  $(D_1/C_1, D_2/C_2)$  where  $C_1$  and  $C_2$  are chunk size factors. Second, we use the chunked weight matrix as importance gates that shrinks the weights close to zero when they are deemed unimportant. We show the CHIM-based representation method in Figure 2.

We start by linearly transforming the concatenated attributes into  $c$ . Then we reshape  $c$  into  $C$  with shape  $(D_1/C_1, D_2/C_2)$ . These operations are similar to Equations 13 and 14. We then repeat this matrix  $C_1 * C_2$  times and concatenate them such that we create a matrix  $W'$  of shape  $(D_1, D_2)$ . Finally, we use the sigmoid function  $\sigma$  to transform the matrix into gates that represent importance:

$$W' = \sigma \begin{bmatrix} C; \dots; C \\ \dots; \dots; \dots \\ C; \dots; C \end{bmatrix} \in [0, 1]^{D_1 \times D_2} \quad (16)$$

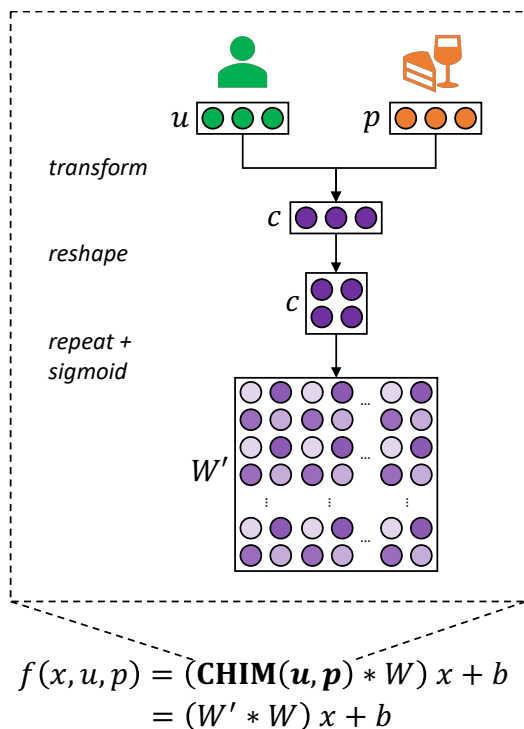


Figure 2: CHIM-based attribute representation and injection to a non-linear function in the model.

Finally we broadcast-multiply  $W'$  with the original weight matrix  $W$  to shrink the weights. The result is a sparse version of  $W$ , which can be seen as either a regularization step (Ng, 2004) where most weights are set close to zero, or a correction step (Amplayo et al., 2018b) where the important gates are used to correct the weights. The use of multiple chunks regards CHIM as coarse-grained access control (Shen et al., 2019) where the use of different important gates for every node is unnecessary and expensive. The final function is shown below:

$$f(x, u, p) = (W' * W)x + b \quad (17)$$

To summarize, chunking helps reduce the number of parameters while retaining the model performance, and importance matrix makes optimization easier during training, resulting to a performance improvement. We also tried alternative methods for importance matrix such as residual addition (i.e.,  $\tanh(W') + W$ ) introduced in He et al. (2016), and low-rank adaptation methods (Jaech and Ostendorf, 2018; Kim et al., 2019), but these did not improve the model performance.

### 2.3 Where: Attribute Injection

Using the approaches described above, we can inject attribute representation into four different

parts of the model. This section describes what it means to inject attributes to a certain location and why previous work have been injecting them in the worst location (i.e., in the attention mechanism).

**In the attention mechanism** Injecting attributes to the attention mechanism means that we bias the selection of more informative words during pooling. For example, in Figure 1b, a user may find delicious drinks to be the most important aspect in a restaurant. Injection in the attention mechanism would bias the selection of words such as *wine*, *smooth*, and *sweet* to create the document encoding. This is the standard location in the model to inject the attributes, and several (Chen et al., 2016; Amplayo et al., 2018a) have shown how the injected attention mechanism selects different words when the given user or product is different.

We argue, however, that attention mechanism is not the best location to inject the attributes. This is because we cannot obtain user- or product-biased sentiment information from the representation. In the example above, although we may be able to select, with user bias, the words *wine* and *sweet* in the text, we do not know whether the user has a positive or negative sentiment towards these words (e.g., Does the user like wine? How about *sweet* wines? etc.). In contrast, the three other locations we discuss below use the attributes to modify how the model looks at sentiment at different levels of textual granularity.

**In the word embedding** Injecting attributes to the word embedding means that we bias the sentiment intensity of a word independent from its neighboring context. For example, if a user normally uses the words *tasty* and *delicious* with a less and more positive intensity, respectively, the corresponding attribute-injected word embeddings would come out less similar, despite both words being synonymous.

**In the BiLSTM encoder** Injecting attributes to the encoder means that we bias the contextualization of words based on their neighbors in the text. For example, if a user likes their cake sweet but their drink with no sugar, the attribute-injected encoder would give a positive signal to the encoding of *sweet* in the text “*the cake was sweet*” and a negative signal in the text “*the drink was sweet*”.

**In the logistic classifier** Injecting attributes to the classifier means that we bias the probability

Datasets	C	#train	#dev	#test	#users	#products	#docs/user	#docs/product
IMDB	10	67,426	8,381	9,112	1,310	1,635	64.82	51.94
Yelp 2013	5	62,522	7,773	8,671	1,631	1,633	48.42	48.36
Yelp 2014	5	183,019	22,745	25,399	4,818	4,194	47.97	55.11

Table 1: Statistics of the datasets used for the Sentiment Classification task.

distribution of sentiment based on the final document encoding. If a user tends to classify the sentiment of reviews about sweet cakes as highly positive, then the model would give a high probability to highly positive sentiment classes for texts such as “*the cake was sweet*”.

### 3 Experiments

#### 3.1 General Setup

We perform experiments on two tasks. The first task is Sentiment Classification, where we are tasked to classify the sentiment of a review text, given additionally the user and product information as attributes. The second task is Attribute Transfer, where we attempt to transfer the attribute encodings learned from the sentiment classification model to solve two other different tasks: (a) Product Category Classification, where we are tasked to classify the category of the product, and (b) Review Headline Generation, where we are tasked to generate the title of the review, given only both the user and product attribute encodings. Datasets, evaluation metrics, and competing models are different for each task and are described in their corresponding sections.

Unless otherwise stated, our models are implemented with the following settings. We set the dimensions of the word, user, and product vectors to 300. We use pre-trained GloVe embeddings<sup>2</sup> (Pennington et al., 2014) to initialize the word vectors. We also set the dimensions of the hidden state of BiLSTM to 300 (i.e., 150 dimensions for each of the forward/backward hidden state). The chunk size factors  $C_1$  and  $C_2$  are both set to 15. We use dropout (Srivastava et al., 2014) on all non-linear connections with a dropout rate of 0.1. We set the batch size to 32. Training is done via stochastic gradient descent over shuffled mini-batches with the Adadelta update rule (Zeiler, 2012) and with  $l_2$  constraint (Hinton et al., 2012) of 3. We perform early stopping using the development set. Training and experiments are done using an NVIDIA

<sup>2</sup><https://nlp.stanford.edu/projects/glove/>

GeForce GTX 1080 Ti graphics card.

#### 3.2 Sentiment Classification

**Datasets and Evaluation** We use the three widely used sentiment classification datasets with user and product information available: IMDB, Yelp 2013, and Yelp 2014 datasets<sup>3</sup>. These datasets are curated by Tang et al. (2015), where they ensured twenty-core for both users and products (i.e., users have at least twenty products and vice versa), split them into train, dev, and test sets with an 8:1:1 ratio, and tokenized and sentence-split using the Stanford CoreNLP (Manning et al., 2014). Dataset statistics are shown in Table 1. Evaluation is done using two metrics: the accuracy which measures the overall sentiment classification performance, and RMSE which measures the divergence between predicted and ground truth classes.

**Comparisons of different attribute representation and injection methods** To conduct a fair comparison among the different methods described in Section 2, we compare these methods when applied to our base model using the development set of the datasets. Specifically, we use a smaller version of our base model (with dimensions set to 64) and incorporate the user and product attributes using nine different approaches: (1) **bias-attention**: the bias-based method injected to the attention mechanism, (2-5) the matrix-based method injected to four different locations (**matrix-embedding**, **matrix-encoder**, **matrix-attention**, **matrix-classifier**), and (6-9) the CHIM-based method injected to four different locations (**CHIM-embedding**, **CHIM-encoder**, **CHIM-attention**, **CHIM-classifier**). We then calculate the accuracy of each approach for all datasets.

Results are shown in Figure 3. The figure shows that bias-attention consistently performs poorly compared to other approaches. As expected, matrix-based representations perform the

<sup>3</sup><https://drive.google.com/open?id=1PxAkMPLFMnFom46FMMXkHeqIxDbA16oy>

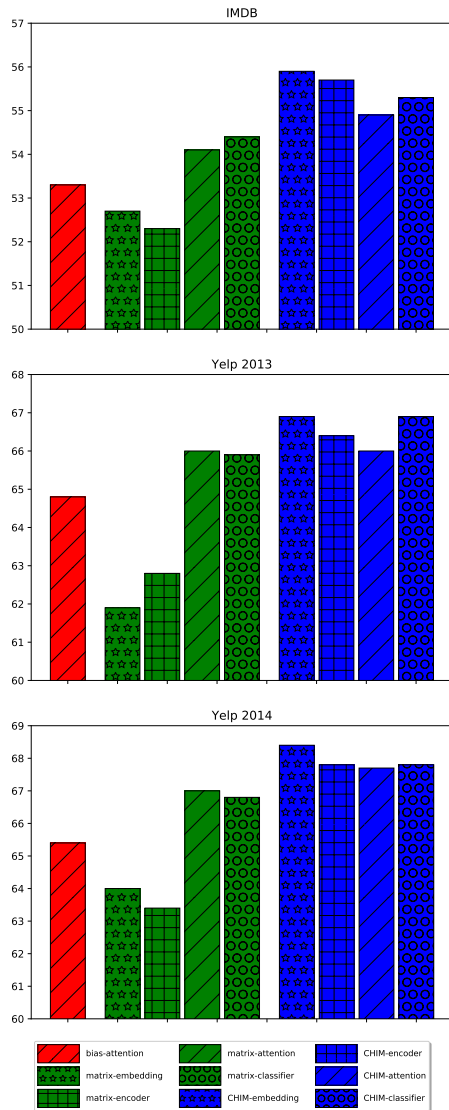


Figure 3: Accuracies (y-axis) of different attribute representation (bias, matrix, CHIM) and injection (emb: embed, enc: encode, att: attend, cls: classify) approaches on the development set of the datasets.

worst when injected to embeddings and encoder, however we can already see improvements over bias-attention when these representations are injected to attention and classifier. This is because the number of parameters used in the the weight matrices of attention and classifier are relatively smaller compared to those of embeddings and encoder, thus they are easier to optimize. The CHIM-based representations perform the best among other approaches, where CHIM-embedding garners the highest accuracy across datasets. Finally, even when using a better representation method, CHIM-attention consistently performs the worst among CHIM-based representations. This shows that attention mechanism is

not the optimal location to inject attributes.

### Comparisons with models in the literature

We also compare with models from previous work, listed below:

1. **UPNN** (Tang et al., 2015) uses a CNN classifier as base model and incorporates attributes as user- and product-specific weight parameters in the word embeddings and logistic classifier.
2. **UPDMN** (Dou, 2017) uses an LSTM classifier as base model and incorporates attributes as a separate deep memory network that uses other related documents as memory.
3. **NSC** (Chen et al., 2016) uses a hierarchical LSTM classifier as base model and incorporates attributes using the bias-attention method on both word- and sentence-level LSTMs.
4. **DUPMN** (Long et al., 2018) also uses a hierarchical LSTM as base model and incorporates attributes as two separate deep memory network, one for each attribute.
5. **PMA** (Zhu and Yang, 2017) is similar to NSC but uses external features such as the ranking preference method of a specific user.
6. **HCSC** (Amplayo et al., 2018a) uses a combination of BiLSTM and CNN as base model, incorporates attributes using the bias-attention method, and also considers the existence of cold start entities.
7. **CMA** (Ma et al., 2017) uses a combination of LSTM and hierarchical attention classifier as base model, incorporates attributes using the bias-attention method, and does this separately for user and product.

Notice that most of these models, especially the later ones, use the bias-attention method to represent and inject attributes, but also employ a more complex model architecture to enjoy a boost in performance.

Results are summarized in Table 2. On all three datasets, our best results outperform all previous models based on accuracy and RMSE. Among our four models, CHIM-embedding performs the best in terms of accuracy, with performance increases of 2.4%, 1.3%, and 1.6% on IMDB, Yelp 2013,

Name	Model		IMDB		Yelp 2013		Yelp 2014	
	Base Model	Injection	Acc	RMSE	Acc	RMSE	Acc	RMSE
UPNN	CNN	embedding, classifier	43.5	1.602	59.6	0.748	60.8	0.764
UPDMN	LSTM	memory networks	46.5	1.351	63.9	0.662	61.3	0.720
NSC	HierLSTM	attention	53.3	1.281	65.0	0.692	66.7	0.654
DUPMN	HierLSTM	memory networks	53.9	1.279	66.2	0.667	<u>67.6</u>	0.639
PMA	HierLSTM	attention <sup>1</sup>	54.0	1.301	65.8	0.668	67.5	0.641
HCSC	BiLSTM+CNN	attention <sup>2</sup>	<u>54.2</u>	1.213	65.7	<u>0.660</u>	–	–
CMA	LSTM+HierAtt	attention <sup>3</sup>	<u>54.0</u>	<u>1.191</u>	<u>66.4</u>	0.677	<u>67.6</u>	<u>0.637</u>
	BiLSTM	embedding	<b>56.4</b>	<b>1.161</b>	<b>67.8</b>	0.646	<b>69.2</b>	0.629
CHIM	BiLSTM	encoder	55.9	1.234	67.0	0.659	68.4	0.631
(Ours)	BiLSTM	attention	54.4	1.219	66.5	0.664	68.5	0.634
	BiLSTM	classifier	55.5	1.219	67.5	<b>0.641</b>	68.9	<b>0.622</b>

Table 2: Sentiment classification results of competing models based on accuracy and RMSE metrics on the three datasets. Underlined values correspond to the best values for each block. **Boldfaced** values correspond to the best values across the board. <sup>1</sup>uses additional external features, <sup>2</sup>uses a method that considers cold-start entities, <sup>3</sup>uses separate bias-attention for user and product.

and Yelp 2014, respectively. CHIM-classifier performs the best in terms of RMSE, outperforming all other models on both Yelp 2013 and 2014 datasets. Among our models, CHIM-attention mechanism performs the worst, which shows similar results to our previous experiment (see Figure 3). We emphasize that our models use a simple BiLSTM as base model, and extensions to the base model (e.g., using multiple hierarchical LSTMs as in Wu et al. 2018), as well as to other aspects (e.g., consideration of cold-start entities as in Amplayo et al. 2018a), are orthogonal to our proposed attribute representation and injection method. Thus, we expect a further increase in performance when these extensions are done.

### 3.3 Attribute Transfer

In this section, we investigate whether it is possible to transfer the attribute encodings, learned from the sentiment classification model, to other tasks: product category classification and review headline generation. The experimental setup is as follows. First, we train a sentiment classification model using an attribute representation and injection method of choice to learn the attribute encodings. Then, we use these fixed encodings as input to the task-specific model.

**Dataset** We collected a new dataset from Amazon<sup>4</sup>, which includes the product category and

<sup>4</sup>[https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon\\_reviews\\_multilingual\\_US\\_v1\\_00.tsv.gz](https://s3.amazonaws.com/amazon-reviews-pds/tsv/amazon_reviews_multilingual_US_v1_00.tsv.gz)

the review headline, aside from the review text, the sentiment score, and the user and product attributes. Following Tang et al. (2015), we ensured that both users and products are twenty-core, split them into train, dev, and test sets with an 8:1:1 ratio, and tokenized and sentence-split the text using Stanford CoreNLP (Manning et al., 2014). The final dataset contains 77,028 data points, with 1,728 users and 1,890 products. This is used as the sentiment classification dataset.

To create the task-specific datasets, we split the dataset again such that no users and no products are seen in at least two different splits. That is, if user  $u$  is found in the train set, then it should not be found in the dev and the test sets. We remove the user-product pairs that do not satisfy this condition. We then append the corresponding product category and review headline for each user-product pair. The final split contains 46,151 training, 711 development, and 840 test instances. It also contains two product categories: Music and Video DVD. The review headline is tokenized using SentencePiece<sup>5</sup> with 10k vocabulary. The datasets are released here for reproducibility: <https://github.com/rktamplayo/CHIM>.

**Evaluation** In this experiment, we compare five different attribute representation and injection methods: (1) the bias-attention method, and (2-5) the CHIM-based representation method in-

<sup>5</sup><https://github.com/google/sentencepiece>

Method	Accuracy	Perplexity
Majority	60.12	–
Random	60.67 $\pm$ 0.27	43.53
Bias - Attention	<b>58.74 <math>\pm</math> 0.49</b>	<b>44.00</b>
CHIM - Embedding	62.26 $\pm$ 0.22	42.71
CHIM - Encoder	64.62 $\pm$ 0.34	42.65
CHIM - Attention	60.95 $\pm$ 0.15	42.78
CHIM - Classifier	61.83 $\pm$ 0.43	42.69

Table 3: Accuracy (higher is better) and perplexity (lower is better) of competing models on the Amazon dataset for the transfer tasks on product category classification and review headline generation, respectively. Accuracy intervals are calculated by running the model 10 times. Performance worse than the random and majority baselines are colored red.

jected to all four different locations in the model. We use the attribute encodings, which are learned from pre-training on the sentiment classification dataset, as input to the transfer tasks, in which they are fixed and not updated during training. As a baseline, we also show results when using encodings of randomly set weights. Moreover, we additionally show the majority class as additional baseline for product category classification.

For the product category classification task, we use a logistic classifier as the classification model and accuracy as the evaluation metric. For the review headline generation task, we use an LSTM decoder as the generation model and perplexity as the evaluation metric.

**Results** For the product category classification task, the results are reported in Table 3. The table shows that representations learned from CHIM-based methods perform better than the random baseline. The best model, CHIM-encoder, achieves an increase of at least 3 points in accuracy compared to the baseline. This means that, interestingly, CHIM-based attribute representations have also learned information about the category of the product. In contrast, representations learned from the bias-attention method are not able to transfer well on this task, leading to worse results compared to the random and majority baseline. Moreover, CHIM-attention performs the worst among CHIM-based models, which further shows the ineffectiveness of injecting attributes to the attention mechanism.

Results for the review headline generation task are also shown in Table 3. The table shows less

promising results, where the best model, CHIM-encoder, achieves a decrease of 0.88 points in perplexity from the random encodings. Although this still means that some information has been transferred, one may argue that the gain is too small to be considered significant. However, it has been well perceived, that using only the user and product attributes to generate text is unreasonable, since we expect the model to generate coherent texts using only two vectors. This impossibility is also reported by Dong et al. (2017) where they also used sentiment information, and Ni and McAuley (2018) where they additionally used learned aspects and a short version of the text to be able to generate well-formed texts. Nevertheless, the results in this experiment agree to the results above regarding injecting attributes to the attention mechanism; bias-attention performs worse than the random baseline, and CHIM-attention performs the worst among CHIM-based models.

### 3.4 Where should attributes be injected?

All our experiments unanimously show that (a) the bias-based attribute representation method is not the most optimal method, and (b) injecting attributes in the attention mechanism results to the worst performance among all locations in the model, regardless of the representation method used. The question “where is the best location to inject attributes?” remains unanswered, since different tasks and settings produce different best models. That is, CHIM-embedding achieves the best accuracy while CHIM-classifier achieves the best RMSE on sentiment classification. Moreover, CHIM-encoder produces the most transferable attribute encoding for both product category classification and review headline generation. The suggestion then is to conduct experiments on all locations and check which one is best for the task at hand.

Finally, we also investigate whether injecting in to more than one location would result to better performance. Specifically, we jointly inject in two different locations at once using CHIM, and do this for all possible pairs of locations. We use the smaller version of our base model and calculate the accuracies of different models using the development set of the Yelp 2013 dataset. Figure 4 shows a heatmap of the accuracies of jointly injected models, as well as singly injected models. Overall, the results are mixed and can be summa-



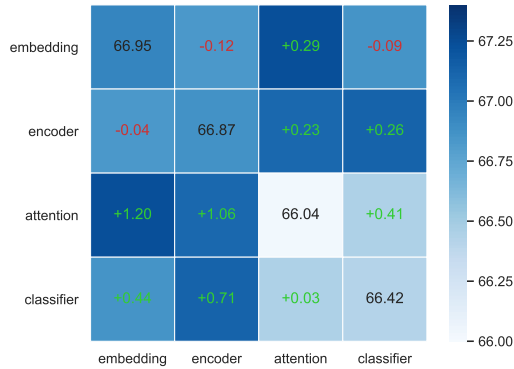


Figure 4: Heatmap of the accuracies of singly and jointly injected CHIM models. Values on each cell represents either the accuracy (for singly injected models) or the difference between the singly and doubly injected models per row.

alized into two statements. Firstly, injecting on the embedding and another location (aside from the attention mechanism) leads to a slight decrease in performance. Secondly and interestingly, injecting on the attention mechanism and another location always leads to the highest increase in performance, where CHIM-attention+embedding performs the best, outperforming CHIM-embedding. This shows that injecting in different locations might capture different information, and we leave this investigation for future work.

## 4 Related Work

### 4.1 Attributes for Sentiment Classification

Aside from user and product information, other attributes have been used for sentiment classification. Location-based (Yang et al., 2017) and time-based (Fukuhara et al., 2007) attributes help contextualize the sentiment geographically and temporally. Latent attributes that are learned from another model have also been employed as additional features, such as latent topics from a topic model (Lin and He, 2009), latent aspects from an aspect extraction model (Jo and Oh, 2011), argumentation features (Wachsmuth et al., 2015), among others. Unfortunately, current benchmark datasets do not include these attributes, thus it is practically impossible to compare and use these attributes in our experiments. Nevertheless, the methods in this paper are not limited to only user and product attributes, but also to these other attributes as well, whenever available.

### 4.2 User/Product Attributes for NLP Tasks

Incorporating user and product attributes to NLP models makes them more personalized and thus user satisfaction can be increased (Baruzzo et al., 2009). Examples of other NLP tasks that use these attributes are text classification (Kim et al., 2019), language modeling (Jaech and Ostendorf, 2018), text generation (Dong et al., 2017; Ni and McAuley, 2018), review summarization (Yang et al., 2018b), machine translation (Michel and Neubig, 2018), and dialogue response generation (Zhang et al., 2017). On these tasks, the usage of the bias-attention method is frequent since it is trivially easy and there have been no attempts to investigate different possible methods for attribute representation and injection. We expect this paper to serve as the first investigatory paper that contradicts to the positive results previous work have seen from the bias-attention method.

## 5 Conclusions

We showed that the current accepted standard for attribute representation and injection, i.e. bias-attention, which incorporates attributes as additional biases in the attention mechanism, is the least effective method. We proposed to represent attributes as chunk-wise importance weight matrices (CHIM) and showed that this representation method significantly outperforms the bias-attention method. Despite using a simple BiLSTM classifier as base model, CHIM significantly outperforms the current state-of-the-art models, even when those models use a more complex base model architecture. Furthermore, we conducted several experiments that conclude that injection to the attention mechanism, no matter which representation method is used, garners the worst performance. This result contradicts previously reported conclusions regarding attribute injection to the attention mechanism. Finally, we show promising results on transferring the attribute representations from sentiment classification, and use them to two different tasks such as product category classification and review headline generation.

## Acknowledgments

We would like to thank the anonymous reviewers for their helpful feedback and suggestions. Reinald Kim Amplayo is grateful to be supported by a Google PhD Fellowship.

## References

- Reinald Kim Amplayo, Jihyeok Kim, Sua Sung, and Seung-won Hwang. 2018a. [Cold-start aware user and product attention for sentiment classification](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2535–2544.
- Reinald Kim Amplayo, Kyungjae Lee, Jinyoung Yeo, and Seung-won Hwang. 2018b. [Translations as additional contexts for sentence classification](#). In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 3955–3961.
- Stefanos Angelidis and Mirella Lapata. 2018. [Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3675–3686.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural machine translation by jointly learning to align and translate](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Andrea Baruzzo, Antonina Dattolo, Nirmala Pudota, and Carlo Tasso. 2009. [A general framework for personalized text classification and annotation](#). In *Proceedings of the Workshop on Adaptation and Personalization for Web 2.0, AP WEB 2.0@UMAP, Trento, Italy, June 22, 2009*.
- Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. [Neural sentiment classification with user and product attention](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1650–1659.
- Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. [Learning to generate product reviews from attributes](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 623–632, Valencia, Spain. Association for Computational Linguistics.
- Zi-Yi Dou. 2017. [Capturing user and product information for document level sentiment analysis with deep memory network](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 521–526. Association for Computational Linguistics.
- Jessica Fidler and Yoav Goldberg. 2017. [Controlling linguistic style aspects in neural language generation](#). *CoRR*, abs/1707.02633.
- Tomohiro Fukuhara, Hiroshi Nakagawa, and Toyooki Nishida. 2007. [Understanding sentiment of people from news articles: Temporal sentiment analysis of social events](#). In *Proceedings of the First International Conference on Weblogs and Social Media, ICWSM 2007, Boulder, Colorado, USA, March 26-28, 2007*.
- Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. 2013. [Modeling user leniency and product popularity for sentiment classification](#). In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 1107–1111.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. [Deep residual learning for image recognition](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778.
- Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. [Improving neural networks by preventing co-adaptation of feature detectors](#). *CoRR*, abs/1207.0580.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long short-term memory](#). *Neural computation*, 9(8):1735–1780.
- Aaron Jaech and Mari Ostendorf. 2018. [Low-rank RNN adaptation for context-aware language modeling](#). *TACL*, 6:497–510.
- Yohan Jo and Alice H. Oh. 2011. [Aspect and sentiment unification model for online review analysis](#). In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 815–824.
- Jihyeok Kim, Reinald Kim Amplayo, Kyungjae Lee, Sua Sung, Minji Seo, and Seung-won Hwang. 2019. [Categorical metadata representation for customized text classification](#). *TACL*, 7:201–215.
- Soo-Min Kim and Eduard H. Hovy. 2004. [Determining the sentiment of opinions](#). In *COLING 2004, 20th International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2004, Geneva, Switzerland*.
- Fangtao Li, Nathan Nan Liu, Hongwei Jin, Kai Zhao, Qiang Yang, and Xiaoyan Zhu. 2011. [Incorporating reviewer and product information for review rating prediction](#). In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 1820–1825.
- Chenghua Lin and Yulan He. 2009. [Joint sentiment/topic model for sentiment analysis](#). In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009*, pages 375–384.

- Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- Yunfei Long, Mingyu Ma, Qin Lu, Rong Xiang, and Chu-Ren Huang. 2018. [Dual memory network model for biased product review classification](#). In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 140–148. Association for Computational Linguistics.
- Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang, and Xu Sun. 2017. [Cascading multiway attentions for document-level sentiment classification](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 634–643. Asian Federation of Natural Language Processing.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. [The stanford corenlp natural language processing toolkit](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60.
- Paul Michel and Graham Neubig. 2018. [Extreme adaptation for personalized neural machine translation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 312–318.
- Andrew Y. Ng. 2004. [Feature selection, l1 vs. l2 regularization, and rotational invariance](#). In *Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04*, pages 78–, New York, NY, USA. ACM.
- Jianmo Ni and Julian McAuley. 2018. [Personalized review generation by expanding phrases and attending on aspect-aware representations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 706–711.
- Bo Pang and Lillian Lee. 2007. [Opinion mining and sentiment analysis](#). *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.
- Dae Hoon Park, Hyun Duk Kim, ChengXiang Zhai, and Lifan Guo. 2015. [Retrieval of relevant opinion sentences for new products](#). In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 393–402.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.
- Yikang Shen, Shawn Tan, Alessandro Sordani, and Aaron C. Courville. 2019. [Ordered neurons: Integrating tree structures into recurrent neural networks](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. [Dropout: a simple way to prevent neural networks from overfitting](#). *Journal of Machine Learning Research*, 15(1):1929–1958.
- Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. [User-level sentiment analysis incorporating social networks](#). In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1397–1405.
- Duyu Tang, Bing Qin, and Ting Liu. 2015. [Learning semantic representations of users and products for document level sentiment classification](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1014–1023.
- Henning Wachsmuth, Johannes Kiesel, and Benno Stein. 2015. [Sentiment flow - A general model of web review argumentation](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 601–611.
- Zhen Wu, Xin-Yu Dai, Cunyan Yin, Shujian Huang, and Jiajun Chen. 2018. [Improving review representations with user attention and product attention for sentiment classification](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5989–5996.
- Min Yang, Jincheng Mei, Heng Ji, Wei Zhao, Zhou Zhao, and Xiaojun Chen. 2017. [Identifying and tracking sentiments and topics from social media texts during natural disasters](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 527–533.
- Min Yang, Qiang Qu, Jia Zhu, Ying Shen, and Zhou Zhao. 2018a. [Cross-domain aspect/sentiment-aware abstractive review summarization](#). In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*,

Torino, Italy, October 22-26, 2018, pages 1531–1534.

Min Yang, Wenting Tu, Qiang Qu, Zhou Zhao, Xiaojun Chen, and Jia Zhu. 2018b. [Personalized response generation by dual-learning based domain adaptation](#). *Neural Networks*, 103:72–82.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. [Hierarchical attention networks for document classification](#). In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489.

Matthew D. Zeiler. 2012. [ADADELTA: an adaptive learning rate method](#). *CoRR*, abs/1212.5701.

Weinan Zhang, Ting Liu, Yifa Wang, and Qingfu Zhu. 2017. [Neural personalized response generation as domain adaptation](#). *CoRR*, abs/1701.02073.

Pengcheng Zhu and Yujiu Yang. 2017. [Parallel multi-feature attention on neural sentiment classification](#). In *Proceedings of the Eighth International Symposium on Information and Communication Technology, Nha Trang City, Viet Nam, December 7-8, 2017*, pages 181–188.