Improving Document-Level Sentiment Analysis with User and Product Context

Chenyang Lyu
School of Computing
Dublin City University
Dublin, Ireland
chenyang.lyu2@mail.dcu.ie

Jennifer Foster
School of Computing
Dublin City University
Dublin, Ireland
jennifer.foster@dcu.ie

Yvette Graham
School of Computer Science
& Statistics
Trinity College Dublin
Dublin, Ireland
ygraham@tcd.ie

Abstract

Past work that improves document-level sentiment analysis by encoding user and product information has been limited to considering only the text of the current review. We investigate incorporating additional review text available at the time of sentiment prediction that may prove meaningful for guiding prediction. Firstly, we incorporate all available historical review text belonging to the author of the review in question. Secondly, we investigate the inclusion of historical reviews associated with the current product (written by other users). We achieve this by explicitly storing representations of reviews written by the same user and about the same product and force the model to *memorize* all reviews for one particular user and product. Additionally, we drop the hierarchical architecture used in previous work to enable words in the text to directly attend to each other. Experiment results on IMDB, Yelp 2013 and Yelp 2014 datasets show improvement to state-of-the-art of more than 2 percentage points in the best case.

1 Introduction

Document-level sentiment analysis aims to predict sentiment polarity of text that often takes the form of product or service reviews. Tang et al. (2015) demonstrated that modelling the individual who has written the review, as well as the product being reviewed, is worthwhile for polarity prediction, and this has led to exploratory work on how best to combine review text with user/product information in a neural architecture (Chen et al., 2016; Ma et al., 2017; Dou, 2017; Long et al., 2018; Amplayo, 2019; Amplayo et al., 2018). A feature common amongst past studies is that user and product IDs are modelled as embedding vectors whose parameters are learned during training. We take this idea a step further and represent users and products using the *text of all the reviews belonging to a single user or product* – see Fig. 1 (left).

There are two reasons to incorporate review text into user/product modelling. Firstly, the reviews from a given user will reflect their word choices when conveying sentiment. For example, a typical user might use words such as *fantastic* or *excellent* with correspondingly high ratings but another user could use the same words sarcastically with a low rating. Similarly, a group of users writing a review of the same product may use the same or similar opinionated words to refer to that product. Secondly, learning meaningful user and product embeddings that are only updated by back propagation is difficult when a user or product only has a small number of reviews, whereas one may still be able to glean something useful from the text of even a small number of reviews.

A naive approach might compute representations of all the reviews of a given user or product each time we have a new training sample but this would be too expensive, and we instead propose the following incremental approach: With each new training sample, we obtain the review text representation, with BERT (Devlin et al., 2019) as our encoder, before using the representation together with user and product vectors to obtain a user-biased document representation and a product-biased document representation, which are then employed to obtain sentiment polarity. We then add the user-biased and product-biased

Proceedings of the 28th International Conference on Computational Linguistics, pages 6724–6729
Barcelona, Spain (Online), December 8-13, 2020

document representations to the corresponding user and product vectors, so that they are ready for the next sample. In doing so, we incrementally store and update representations of reviews for a given user and product. Unlike Ma et al. (2017), who use a hierarchical structure in which sentence representations are first computed before being combined into a document representation, we let the words in the text directly attend to each other. The architecture we propose is depicted on the righthand side of Fig. 1 and is explained in more detail in Section 2.

We compare performance with a range of systems and results show that our approach works, improving on state-of-the-art results for all three benchmark datasets (IMDB, Yelp-13 and Yelp-14).[1] We also compare to a version of our own system which does not use the review text representations to encode user and product information. While it performs competitively with other systems, demonstrating the efficacy of our basic architecture, it does not work as well as our proposed system, particularly for reviews written by users or products with only a small number of reviews.

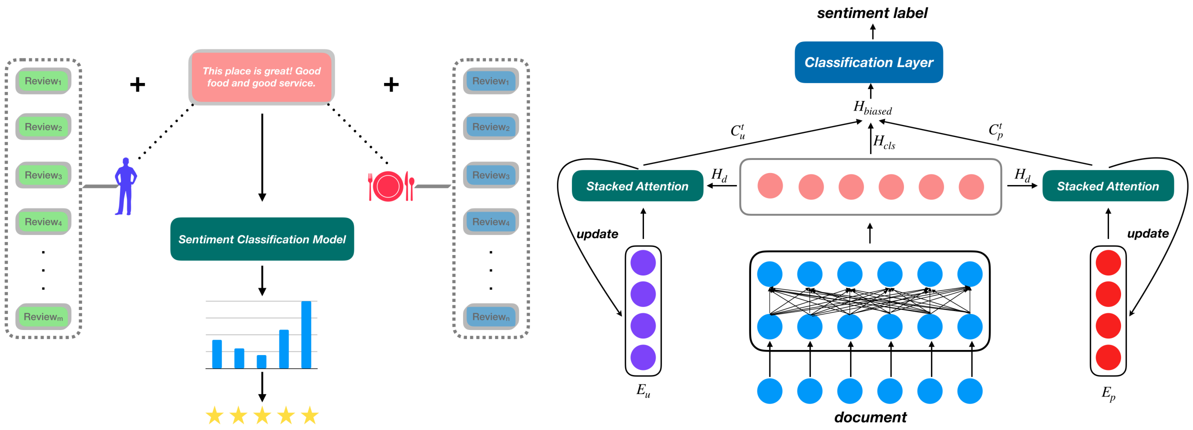

Figure 1: Utilizing all historical reviews of corresponding user and products (left); overall architecture of our model, where E_u and E_p are user and product representations (right).

2 Methodology

An overview of our model architecture is shown in Figure 1 (right). The input to our model consists of d, u, p, which are the document, the user id and the product id respectively. u and p are both mapped to embedding vectors, $E_u, E_p \in \mathbf{R^h}$. d is fed into the BERT encoder to generate a document representation $H_d \in \mathbf{R^{L \times h}}$ where L is the length of document after tokenization. We then inject E_u and E_p, to get the user-product biased document representation $H_{biased} \in \mathbf{R^h}$. Finally, we feed the biased document representation H_{biased} into a linear layer followed by a *softmax* layer to get the distribution of the sentiment label y. We use *cross-entropy* to calculate the loss between the predictions and ground-truth labels.

Injecting user and product preferences We adopt stacked **multi-head-attention** (Q, K, V) (Vaswani et al., 2017) to model the connections between the current document and user/product vectors, which in this work correspond to all historical reviews composed by the user or about the product to date. In a typical dot-product attention (Q, K, V), $Q \in R^{L_Q \times h}$, $K \in R^{L_K \times h}$, $V \in R^{L_V \times h}$. Generally, $L_K = L_v$. E_u and E_p are regarded as queries, H_d as keys and values. We compute the user-specific document representation, C_u^t, and product-specific document representation, C_p^t as follows:

$$C_u^t = \textbf{stacked-attention}(E_u, H_d, H_d) \quad C_p^t = \textbf{stacked-attention}(E_p, H_d, H_d) \tag{1}$$

where $C_u^t = attention(C_u^{t-1})$, $C_u^0 = E_u$ (similarly for C_p^t), and t is the number of layers of the attention function. In Equation (1), $C_u^t \in R^h, C_p^t \in R^h$.

[1] http://ir.hit.edu.cn/˜dytang/paper/acl2015/dataset.7z

We adopt a *gating mechanism* to obtain importance vectors, z_u and z_p, to control the *contribution* of user-specific and product-specific document representations to the output classification:

$$z_u = \sigma(W_{zu}C_u^t + W_{zh}H_d + b_u) \quad z_p = \sigma(W_{zp}C_p^t + W_{zh}H_d + b_p) \tag{2}$$

Finally, we obtain the biased document representation H_{biased} by:

$$H_{biased} = H_{cls} + z_u \odot C_u^t + z_p \odot C_p^t \tag{3}$$

where $H_{cls} \in \mathbf{R^h}$ is the final hidden vector of the [CLS] token (Devlin et al., 2019) and $\odot$ is element-wise product.

Updating the user and product matrix To implement our idea of using all reviews composed by u and all reviews about p, we incrementally add the current user/product-specific document representation to the corresponding entries in the embedding matrix at each step during training:

$$E_u^{'} = \sigma(E_u + \lambda_u C_u^t) \quad E_p^{'} = \sigma(E_p + \lambda_p C_p^t) \tag{4}$$

where λ_u and λ_p are both learnable real numbers that control the degree to which the representation of the current document should be employed.

3 Experiments

3.1 Experimental Setup

Our experiments are conducted on the IMDB, Yelp-13 and Yelp-14 benchmark datasets, statistics of which are shown in Table 1. We use the BERT-base model from HuggingFace (Wolf et al., 2019). We train our model with a learning rate chosen from $\{$8e-6, 3e-5, 5e-5$\}$, and a weight decay rate chosen from $\{$0, 1e-1, 1e-2, 1e-3$\}$, the optimizer we use is AdamW(Loshchilov and Hutter, 2019). In our experiments, the number of attention layers t is set to 5. The maximum sequence length to BERT is 512. We select the hyper-parameters achieving the best results on the dev set for evaluation on the test set. Evaluation metrics (Accuracy and RMSE) are calculated using scripts from Scikit-learn (Pedregosa et al., 2011).[2]

Datasets	Classes	Documents	Users	Products	Docs/User	Docs/Product	Words/Doc
IMDB	1–10	84,919	1,310	1,635	64.82	51.94	394.6
Yelp-2013	1–5	78,966	1,631	1,633	48.42	48.36	189.3
Yelp-2014	1–5	231,163	4,818	4,194	47.97	55.11	196.9

Table 1: Statistics of IMDB, Yelp-2013 and Yelp-2014.

3.2 Results

Our experimental results are shown in Table 2. Our proposed model is named IUPC (**I**ncorporating **U**ser-**P**roduct **C**ontext). The first two rows are baseline models: BERT VANILLA which is the basic BERT model without user and product information, i.e. only review text, and IUPC W/O UPDATE, which is the same as our proposed model except that we do not update the user and product embedding matrix by incrementally adding the new review representations. The third row shows our proposed model. We also compare with results from the NLP-progress leaderboard[3] of the following models:

[2] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics

[3] http://nlpprogress.com/english/sentiment_analysis.html

CHIM (Amplayo, 2019) adopts a chunk-wise matrix representation for user/product attributes; injects user/product information in different locations.

CMA (Ma et al., 2017) A hierarchical LSTM encoding the document; injects user and product information hierarchically.

DUPMN (Long et al., 2018) encodes the document using a hierarchical LSTM; adopts two memory networks, one for user information and another for product information.

HCSC (Amplayo et al., 2018) A combination of CNN and Bi-LSTM as the document encoder; injects user/product information with bias-attention.

HUAPA (Wu et al., 2018) adopts two hierarchical models to get user and product specific document representations respectively.

NSC (Chen et al., 2016) A hierarchical LSTM encoder incorporating user/ product attributes with word and sentence-level attention.

RRP-UPM (Yuan et al., 2019) uses two memory networks besides the user/product embeddings to get refined representations for user/product information.

UPDMN (Dou, 2017) An LSTM model encoding the document; a memory network capturing user/product information.

UPNN (Tang et al., 2015) adopts a CNN-based encoder and injects user/product information in the embedding and classification layers.

	IMDB		Yelp-2013		Yelp-2014	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
BERT VANILLA	$47.9_{0.46}$	$1.243_{0.019}$	$67.2_{0.46}$	$0.647_{0.011}$	$67.5_{0.71}$	$0.621_{0.012}$
IUPC W/O UPDATE	$52.1_{0.31}$	$1.194_{0.010}$	$69.7_{0.37}$	$0.605_{0.007}$	$70.0_{0.29}$	$0.601_{0.007}$
IUPC (our model)	$53.8_{0.57}$	$\mathbf{1.151_{0.013}}$	$\mathbf{70.5_{0.29}}$	$\mathbf{0.589_{0.004}}$	$\mathbf{71.2_{0.26}}$	$\mathbf{0.592_{0.008}}$
UPNN	43.5	1.602	59.6	0.784	60.8	0.764
UPDMN	46.5	1.351	63.9	0.662	61.3	0.720
NSC	53.3	1.281	65.0	0.692	66.7	0.654
CMA	54.0	1.191	66.3	0.677	67.6	0.637
DUPMN	53.9	1.279	66.2	0.667	67.6	0.639
HCSC	54.2	1.213	65.7	0.660	67.6	0.639
HUAPA	55.0	1.185	68.3	0.628	68.6	0.626
CHIM	**56.4**	1.161	67.8	0.641	69.2	0.622
RRP-UPM	56.2	1.174	69.0	0.629	69.1	0.621

Table 2: Experimental Results on IMDB, Yelp-2013 and Yelp-2014. Following previous work, we use Accuracy (Acc.) and Root Mean Square Error (RMSE) for evaluation. There are 10 classes in IMDB and 5 classes in Yelp 2013 and Yelp 2014. We run BERT VANILLA, IUPC W/O UPDATE and IUPC five times and report the average Accuracy and RMSE. The subscripts represent standard deviation.

Our model achieves the best classification accuracy and RMSE on Yelp-2013 and Yelp-2014, and the best RMSE on IMDB. It outperforms previous state-of-the-art results by 1.5 accuracy and 0.042 RMSE on Yelp-2013, by 2.1 accuracy and 0.029 RMSE on Yelp-2014, and by 0.01 RMSE on IMDB. Moreover, it outperforms the two baselines, BERT VANILLA and IUPC W/O UPDATE in both classification accuracy and RMSE on all three datasets. Although the classification accuracy of our model on IMDB is lower than most of the previous models, we suspect this is because the BERT model is not good at handling longer documents since the input length to BERT is fixed and the average length of documents in IMDB dataset is much longer than the other two datasets. However, it is worth noting that our model achieves the lowest RMSE which means the predictions of our model are *closer* to the gold labels.

3.3 Analysis

We analyse the results for reviews whose user or product do not have many reviews in the training set and compare our model's performance to the IUPC W/O UPDATE baseline for one dataset (Yelp-2013 dev).

We select only reviews where the number of reviews by that user or for that product falls below three thresholds: 40%, 60%, 80%, where % stands for the number of reviews for a given user/product relative to the average number of reviews for all users/products. Table 3 shows that our model performs better than IUPC W/O UPDATE when there are only a small number of previous reviews available for a given product/user. In other words, when a user or product does not have many reviews, its IUPC W/O UPDATE embedding which is only updated by gradient descent, cannot capture user/product preference as well as our model which explicitly takes advantage of historical review text in its user/product representations.

	40%		60%		80%	
	Acc. (%)	RMSE	Acc. (%)	RMSE	Acc. (%)	RMSE
IUPC W/O UPDATE	63.0	0.608	64.0	0.665	66.8	0.643
IUPC (our model)	**65.7**	**0.585**	**66.8**	**0.649**	**67.9**	**0.631**

Table 3: Analysis of three lower-resource scenarios where % denotes a threshold filter corresponding to the proportion of reviews available relative to the average number in the dataset Yelp-2013 (dev).

In order to get a better idea of where there is room for improvement for IUPC, we examine the 43 Yelp-13 dev set cases, where the predicted label differs from the gold label by more than two points. There are a handful of cases of sarcasm, e.g. *that **lovely** tempe waste/tap water taste in the food*, but the most noteworthy phenomenon is mixed sentiment, e.g. *tacos were good the soup was not tasty*, or the more subtle *brave the scary parking and lack of ambiance*. It is not always clear from the reviews which aspect of the service the rating is directed towards. This suggests that aspect-based sentiment analysis (Pontiki et al., 2014) might be useful here, and training an IUPC model for this task is a possible avenue for future work.

4 Conclusion

In this paper, we propose a neural sentiment analysis architecture that explicitly utilizes all past reviews from a given user or product to improve sentiment polarity classification on the document level. Our experimental results on the IMDB, Yelp-13 and Yelp-14 datasets demonstrate that incorporating this additional context is effective, particularly for the Yelp datasets. The code used to run the experiments is available for use by the research community.[4]

Acknowledgements

This work was funded by Science Foundation Ireland through the SFI Centre for Research Training in Machine Learning (18/CRT/6183). We also thank the reviewers for their helpful comments.

References

Reinald Kim Amplayo, Jihyeok Kim, Sua Sung, and Seung-won Hwang. 2018. Cold-start aware user and product attention for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2535–2544, Melbourne, Australia, July. Association for Computational Linguistics.

Reinald Kim Amplayo. 2019. Rethinking attribute representation and injection for sentiment classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5602–5613, Hong Kong, China, November. Association for Computational Linguistics.

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*, pages 1650–1659, Austin, Texas, November. Association for Computational Linguistics.

[4] https://github.com/lyuchenyang/Document-level-Sentiment-Analysis-with-User-and-Product-Context

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. 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, Copenhagen, Denmark, September. Association for Computational Linguistics.

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, Brussels, Belgium, October. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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, Taipei, Taiwan, November. Asian Federation of Natural Language Processing.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 1014–1023, Beijing, China, July. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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. *CoRR*, abs/1801.07861.

Zhigang Yuan, Fangzhao Wu, Junxin Liu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2019. Neural review rating prediction with user and product memory. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7,2019*, pages 2341–2344.