User and Product-aware Sentiment Classification

Ankitkumar Patel and Tejinderpal Singh
Final Project - W266 NLP

Abstract

Sentiment classification is a fundamental problem which aims to extract user's overall sentiments about products from their reviews. The sentiment classification problem has been exhaustively studied since last decade where the proposed solutions primarily focus on local text of reviews to capture sentiments. However, the sentiments of a review depend not only on the context captured in local text, but also on other exogenous factors like who wrote the review and what product the review is written for. In this paper, we consider user preferences and product characteristics in addition to local text to improve sentiment classification. The challenge is how to incorporate user and product information along with the local text of a review for sentiment analysis. We propose a novel approach of enhancing Glove word embedding with user and product information and applied the user-product specific embedding to train a hierarchical Bi-LSTM network with attention. The proposed solution is compared with a baseline model where user and product information is ignored from the hierarchical Bi-LSTM network. The performance of the proposed model is evaluated on the IMDB reviews while considering only user or product information. The proposed model improves the prediction accuracy compared to the baseline approach by 20% when only user information is considered and 10% when only product information is considered at the embedding layer. The performance can be further improved if both user and product information is considered at the embedding layer.

Keywords: Sentiment, Embedding, Bi-LSTM, Attention, Neural network

1. Introduction

Sentiment classification is a fundamental problem in the field of opinion mining and natural language processing. Recently, sentiment analysis draws increasing attention as the rapid growth of online reviews enables offering personalized recommendations. The task is to infer the sentiment polarity or intensity of a review. Dominating studies regard the problem as a multi-class classification problem [1, 2] and most of these studies focus on designing machine learning algorithms and effective feature designing using neural networks [3, 4].

Most proposed models consider context captured in the local text of a review as input and generate the semantic representations using well-designed neural networks. However, these methods ignore other crucial characteristics like user preferences and product characteristics that can also be used as additional context and may significantly influence on the assigned polarity. For example, a critical reviewer may express "It's a good product" in a review and give the product a 3 star rating, while a lenient reviewer may assign a 5 star rating for an identical review. Thus, user preference affects the sentiment rating of a review. Similarly, product quality may also influence review rating. For example, the features of a high quality product (e.g. i-Phone) tend to receive high ratings compared to that of a low quality product. Therefore, a smart sentiment classifier can be realized using additional context like user preferences and product characteristics in addition to the context from local text.

In this paper, we propose a user and product-aware sentiment classifier. The classifier takes as input a variable-sized review as well as the ID of a user who wrote the review and the ID of a product which is evaluated. The classifier outputs the probability distribution of selecting a polarity. We encode users and products in a multi-dimensional space as real vectors using the word2vec algorithm and integrate these encodings with the pre-generated Glove word encoding to generate user and product aware representation of reviews. These enhanced representations of reviews are input to the two-layer hierarchical Bi-LSTM network to extract out sentiments from words and sentences. To address the memory-loss issue, an attention layer is augmented at each Bi-LSTM layer. We evaluate the performance of the proposed model with a baseline model in which user and product information is ignored from the model. We consider the IMDB movie review dataset to test the efficacy of the model, and demonstrate that the proposed user and product-aware model outperforms the baseline model by at least 10%.

To summarize, our effort provide the following three contributions:

(1) We propose an effective sentiment classification technique by considering additional contexts, such as user preferences and product characteristics in the embedding layer.

(2) We apply the word2vec algorithm to obtain the encoding of each user and product in a multidimensional space

(3) We design a sentiment classifier using hierarchical word and sentence-level Bi-LSTM layers. Next, the classifier is trained using the enhanced embedding which encodes not only similarity and context between words, but also user preferences and product characteristics.

(4) We conduct experiments on IMDB movie review dataset to verify the effectiveness of our model.

The rest of the paper is organized as follows. Section 2 gives a quick overview on the prior art of the problem. In Section 3, we describe the formal problem statement of sentiment classification. The architecture of the proposed model is described in Section 4 and the performance of the proposed model is analyzed in Section 5. Finally, the paper is summarized in Section 6.

2. Background

Sentiment classification is a classical NLP problem which targets at inferring the polarity of a review. Pang and Lee [1, 2] formulated this problem as a classification problem and applied supervised learning algorithms to predict polarities. Goldberg and Zhu [5] applied a semi-supervised learning method to infer the polarities of unlabeled reviews. Many other studies focused on extracting effective features [3, 4] to effectively classify sentiments.

Most of these studies primarily focused on extracting features and context from the local text of a review. Until recently when Tang et. al. proposed the first model by incorporating user and product information in convolution neural networks (CNN) [6] and have shown significant improvements in sentiment classification. To capture document level context, Duo et. al. [7] and Long et. al. [8] proposed deep memory networks where user and product information is injected in memory nodes, and recently, Chen et. al. applied user and product information as attention over hierarchical LSTMs and demonstrated superior performance improvements over the prior efforts [9]. Unlike applying user and product information in the attention layer, in this paper, we incorporate user and product information in the embedding layer of the hierarchical Bi-LSTM network.

3. Problem Statement

The sentiment classification aims to extract a user's overall sentiment about a product from a review. User's sentiments depend not only on specific words and sentences used in a review, but also on the contexts in which they are used. These contexts include other words and sentences used in conjunction, who has written the review, and what product has been reviewed. The challenge is how to accurately predict the polarity of a review using local context, user, and product information. The problem is formally defined as follows.

We are given a set of reviews $R = \{r_1, r_2, r_3, \ldots r_m\}$, sentiment $q_i \in Q$ for each review r_i, ID u_i of a user who wrote review r_i, and ID p_i of a product on which review r_i is written, where Q is a set of sentiment categories. We represent a review as a document d with n sentences $\{s_1, s_2, s_3, \ldots s_n\}$, and l_i is the length of i^{th} sentence. The i^{th} sentence s_i consists of l_i words as $\{w_1, w_2, w_3, \ldots w_{l_i}\}$. The sentiment classification aims to predict the sentiment distributions or ratings of the reviews using the local text, user, and product information.

4. User and Product-Aware Sentiment Classifier

We model sentiment representation using Long Short-Term Memory (LSTM) network because of its excellent performance on sentiment classification for long sentences. To capture word-level and sentence-level contexts, a hierarchical architecture of word and sentence level LSTM networks is considered as shown in Fig. 1. To address the vanishing gradient issue and capture contexts from long sentences and reviews, attention is applied at both, word and sentence layers.

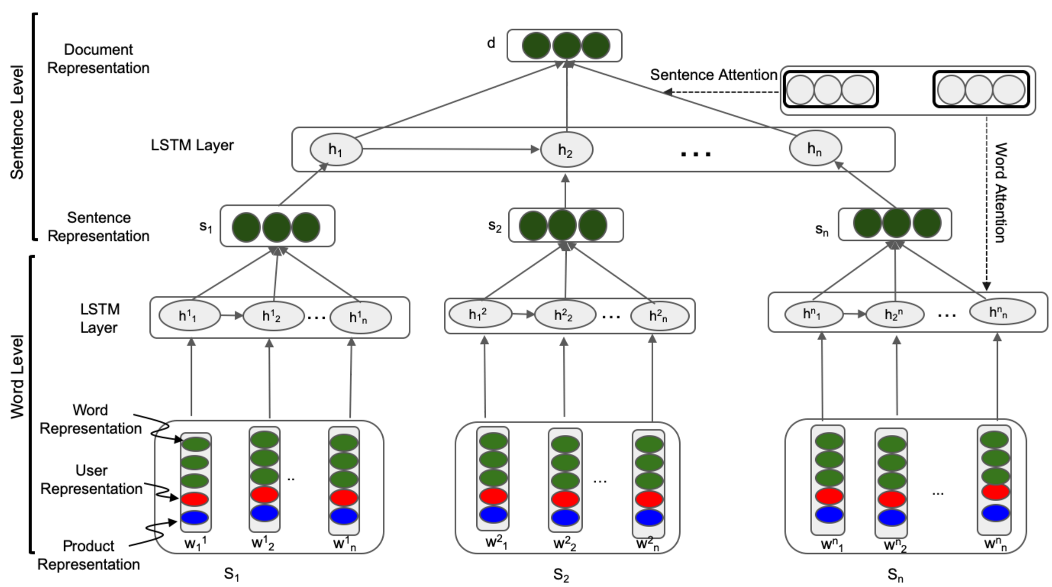

Figure 1. User and product aware sentiment classification architecture

The first-layer is a word-level Bi-LSTM network. The number of LSTM nodes at the first layer is equal to the maximum allowable words in a sentence. Input to the first layer is a sequence of words in a sentence, where each word is represented as a real-valued vector. The vector representation of a word is obtained through the user and product-aware embedding process which is described in the following subsection. User and product-aware embedding takes into account user preferences and product characteristics in addition to the context in which words are used. Attention is applied to the output of each LSTM node of the first layer before the output is aggregated to get a sentence representation. Similarly, the second-layer is a sentence-level Bi-LSTM network. The number of LSTM nodes at the second layer is equal to the maximum allowable sentences in a review. Just like the first layer, attention is applied at the output of the second layer before feeding the output to the softmax layer. The output of the softmax layer is equal to the number of possible polarities that can be assigned to a review.

The enhanced embedding, attention mechanism, and sentiment classification procedures are described in detail in the following subsections.

4.1 Enhanced Embedding

Word embedding is the encoding of a word as a real-valued vector in a multi-dimensional space, which can be obtained by applying the Word2vec or Glove algorithm [10] on a large corpus of data. This embedding process captures similarity among words which are used in similar contexts. Such word embedding is precomputed over the large corpus of Internet data; however, the embedding does not capture user preferences and product characteristics. We redefine the word embedding by augmenting user preference and product characteristics as real-vectors in multidimensional space. A user tends to use a similar set of words in a review while assigning a polarity. User embedding encodes these words such that the cosine similarities between the words used for a polarity is minimum for a user. Similarly, product embedding encodes words such that the cosine similarities between words used for a polarity is minimum for a product.

Let, $w_j^{\,i}$ denotes the user and product-aware word embedding of word j in sentence i, where $w_j^{\,i} \in R^d$. This enhanced embedding is obtained by the concatenation of $u_j^{\,n}$ and $p_j^{\,m}$ to r_j ; $r_j \oplus u_j^n \oplus p_j^m$, where $u_j^{\,n} \in R^{d_u}$ represents word j for user n, $p_j^{\,m} \in R^{d_p}$ represents word j for product m, and $r_j \in R^{d_r}$ represents word j based on the context in which the word has been used in the large corpus of data.

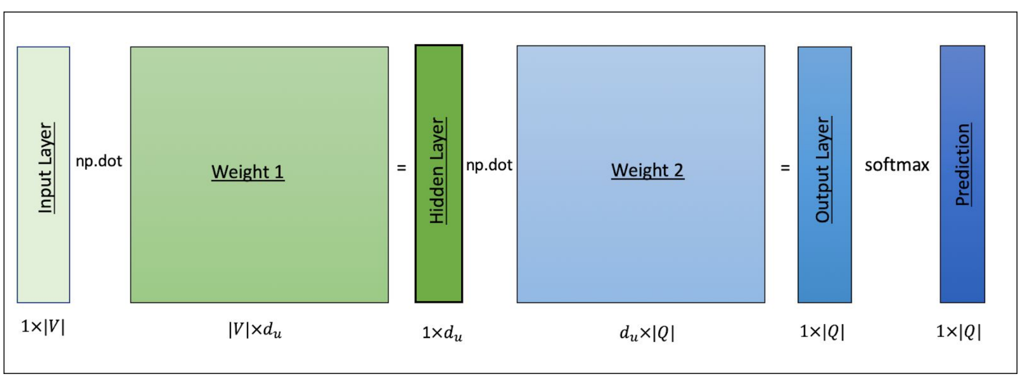

Figure 2. Neural network architecture used in user and product-level embedding

Let's V be the set of vocabularies used in a dataset. $u_j^{\,n} \;\; \forall j \in V$ is obtained by applying the Word2vec algorithm on the reviews of each user n. In the Word2vec algorithm, one hot encodings of words in a review are considered as input and the polarity of the review is considered as a labeled output in a two-layer neural network as shown in Fig. 2. The number of nodes at the first hidden layer is the same as the embedding dimension of user, d_u. The number of nodes at the output layer is the same as the number of polarities. The output of the second hidden layer is fed to the softmax layer. The size of the weight matrix W_1 at the first hidden layer is $|V| \times d_u$. The neural network is trained using the reviews from a user and the weight matrix W_1 represents the embedding of words for the user while considering the user's preferences. Similar process is applied to calculate product embedding for each product while considering product characteristics.

4.2 Attention

The sentiment of a word depends on the context in which the word is used in a sentence. The context is captured by other words in the sentence and these words may be present at an arbitrary long distance from the target word. To capture attention from such context words, we apply the attention mechanism at word-level and sentence-level LSTM networks.

At the word-level attention layer, instead of feeding hidden state outputs from the word-layer LSTM layer to the sentence-layer LSTM layer after average pooling, the hidden layer output is first amplified by amounts in proportion to the importance of other context words that are extracted through attention mechanism. Finally, the weighted sum of hidden states are fed to the sentence-layer LSTM layer. Thus, sentence s_i is represented as follows.

$$s_i = \sum_{j=1}^{l_i} \alpha_j^{\,i} h_j^{\,i}$$

Here, $h_j^{\,i}$ is the hidden layer for j^{th} word of sentence i. $\alpha_j^{\,i}$ denotes the importance of word j in sentence i, which can be derived using the attention mechanism as follows.

$$\alpha_j^{\,i} = \frac{exp(f(h_j^{\,i}))}{\sum_{k=1}^{l_i} exp(f(h_k^{\,i}))}$$

f is a score function that captures the importance of context words in a sentence. f is defined as follows.

$$f(h_j^{\,i}) = tanh(W_h h_j^{\,i})$$

W_h denotes the weight matrix which is configured during the model training period. Sentence-level attention process captures the importances of other sentences for a target sentence and the procedure is identical to the word-level attention procedure.

4.3 Sentiment Classification

Finally, the combined output of the sentence-level LSTM nodes after applying attention captures the representation of a review k highlighting the sentiments while considering user and product information. We can regard k as the features of a review and apply a nonlinear function to map these features into the number of distinct classes C which represents polarities.

$$g_c = tanh(W_c k + b_c)$$

Next, we apply a softmax function to get the distribution of probabilities in the set of distinct classes.

$$q_c = \frac{exp(g_c)}{\sum_{c \in C} exp(g_c)}$$

Here, q_c represents the probability of being in polarity class c. In the model, cross-entropy error between true sentiments and the sentiment distributions derived from the model is define as loss function L for optimization.

$$L = -\sum_{r \in R} \sum_{c \in C} p_c(r)\, q_c(r)$$

Where, $p_c(r)$ denotes the true polarity assigned to review r and $q_c(r)$ denotes the probability of selecting class c for a review r.

5. Performance Analysis

We evaluate the performance of the proposed user and product-aware sentiment classifier with the baseline approach. In the baseline approach, we consider the hierarchical word and sentence-level Bi-LSTM network with attention; however, instead of feeding the network with user and product-aware enhanced word embedding, only the Globe word embedding is fed to the network. We evaluate the effectiveness of the model using the IMDB classification datasets with user and product information, which is built by Tang et al. [6, 11]. The dataset contains 67000 training reviews and 8000 test reviews with user ID, product ID, review text, and sentiment information. Average number of reviews per user is 370 with standard deviation of 61, and similarly, the average number of reviews per product is 41 with standard deviation of 30. Sentiment classes are between 1 and 10 inclusively and the vocabulary size is approximately 80000. We use accuracy as a measure of effectiveness which is defined as the ratio of number of correctly estimated reviews to the total number of reviews. We use the pre-trained 50-dimensional Glove word embeddings. We set the user and product embedding dimension to be 10 each. User and product embeddings are obtained by training the two-layer neural network for each user and product offline. We consider 50 Bi-LSTM nodes at the word level LSTM layer and 15 Bi-LSTM nodes at the sentence level LSTM layer.

Dataset	Considered Users	Considered Products	Baseline	Proposed model
25k/67k	180/180	1/1600	16%	36%
5.6k/67k	1/180	300/1600	14%	24%
67k/67k	180/180	1600/1600	16%	TBD

Table 2: Performance of user and product-aware sentiment classifier

Table 2 shows the prediction accuracy of the proposed model while considering only user information, only product information, and both in addition to word embedding. Due to the limited memory resources, we could consider 25k reviews for training purposes while considering only user information. The proposed model improves the classification accuracy by 20% compared to the baseline approach. Similarly, we could only consider 6k reviews for training purposes while considering only product information. The product embedding improves the classification accuracy by 10% compared to the baseline approach. We could not evaluate the performance while considering both user and product information due to the computational and memory resource limitations. However, we believe that considering both user and product embeddings may further enhance the classification accuracy.

6. Summary

In this paper, we address the user and product-aware sentiment classification problem under a supervised learning framework. We propose a novel approach of capturing user preferences and product characteristics in the representation of a review at the embedding layer. The enhanced embeddings obtained by our proposed approach is applied to the hierarchical Bi-LSTM network with attention. The performance of the proposed model is validated against a baseline model where user and product information is ignored at the embedding layer. We tested the efficacy of the model on the IMDB dataset. Compared to the baseline approach, the proposed solution significantly improves the accuracy of estimating sentiments. The source code of the proposed model is accessible through the git repo [12].

References

1. Bo Pang and Lillian Lee. 2005. Seeing stars: Exploit- ing class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115– 124.
2. Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in infor- mation retrieval*, 2(1-2):1–135.
3. Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mo- hammad. 2014. Sentiment analysis of short in- formal texts. *Journal of Artificial Intelligence Re- search*, pages 723–762.
4. Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. In *COLING*, pages 913–921.
5. Andrew B Goldberg and Xiaojin Zhu. 2006. Seeing s- tars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *GraphBased Method for NLP*, pages 45–52.
6. Duyu Tang, Bing Qin, and Ting Liu, "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)*. volume 1, pages 1014–1023, 2015.
7. Zi-Yi Dou, "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, 2017.
8. Yunfei Long, Mingyu Ma, Qin Lu, Rong Xiang, and Chu-Ren Haung, "Dual Memory Network Model for Biased Product Review Classification", *In Proceeding of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment, and Social Media Analysis*. Pages 140-148, 2018.
9. Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu, "Neural sentiment classification with user and product attention", *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1650–1659, 2016.
10. Jeffery Pennington, Richard Socher, and Christopher Manning, "Glove: Global Vectors for Word Representation," *In Proceeding of the 2014 Conference on Empirical Methods on Natural Language Processing*, Pages 1532-1543, Oct 2014.
11. Data Source for Amazon, Yelp, and Moview Reviews available at - https://drive.google.com/drive/folders/1pcyaXMTx6UAN3mNGUuXsfx0zEffy6IPG
12. User and product-aware sentiment classification source code available at - https://github.com/dr-patel-an/final_project_NLP