Document-Level Multi-Aspect Sentiment Classification with Hierarchical Neural Network

Xi Chen
Department of Computer Science
Princeton University
xc11@princeton.edu

Abstract

Document-level multi-aspect sentiment classification refers to problems concerning recognition of emotions in long text regarding certain aspects. The solution to this problem can have wide applications in real world, such as user preferences predictions and public opinions analysis. To address this problem, a hierarchical neural network model with hop mechanism is introduced. This model chunks input document into small sentence sequences and encodes information in a word-sentence-document order using bi-directional LSTMs. Experiments conducted on BeerAdvocate dataset have shown significant improvements on accuracies of our model over the classic baseline model in most aspects.

1 Introduction

Multi-aspect sentiment classification is a fundamental task in the area of sentiment classification, and it generally refers to problems concerning recognition of emotion(s)/opinions in text regarding certain aspects (Liu et al., 2005). The main advantage of it over traditional sentiment classification tasks is the probability to capture different sentiment responses given different object of interests. For example, given a review 'The restaurant has good food, but bad services.', the sentiment for the aspect 'food' is positive while the sentiment for 'service' is negative.

The solution to the multi-aspect sentiment classification task can have wide applications in the real world. A good example is that it can analyze online reviews on websites such as Airbnb, Yelp, Amazon to predict user preferences of certain products or services, which provides important information for future optimizations/improvements or customer relationship management. Another possible application is to analyze public opinions on certain aspects based on contents from social medias such as Twitter or Facebook.

Since user-generated reviews tend to be comprehensive and long documents, the problem of document-level multi-aspect sentiment classification was introduced to the field. The major challenge of performing sentiment classification on the document level comes from the difficulty of encoding the intrinsic semantic or syntactic relations between sentences in the semantic meaning of document (Tang et al., 2015). Current studies have not proposed a very effective solution to the problem, and most of the existing works have achieved only marginal improvements (Tang et al., 2015).

In this paper, I propose a hierarchical neural network with hop mechanism to approach the problem. Based on the experiments on the BeerAdvocate dataset, the model has been proved to give better performance over baseline model in most aspects.

2 Related Works

Early attempts to approach the document-level sentiment classification problem adopted conventional machine learning models such as decision trees, Naive Bayes and SVM (Pang et al., 2002). These methods are straightforward in understanding and easy to implement, but since they are based on the assumption that documents are represented as a flat feature vector, their ability to extract information on the sentence-level and document-level is limited (Yessenalina et al., 2010).

Later, a two-staged model was proposed for the document-level classification, where in the first stage a classifier is used to predict sentiment class of each sentence and in the second stage another classifier is used do the classification of the document's sentiment class based on the output of the first stage (Zhang et al., 2008).

In recent years, with the rapid development of deep learning techniques, the neural network based methods have gained popularity in this field. RNN based models (Tang et al., 2015) and attention based models (Yang et al., 2016) were proposed and had been proved to have state-of-art performance on the document-level sentiment classifications

However, both the two-staged model and the neural network based models mentioned above output only a single overall sentiment value for each document, which is not sufficient for multi-aspect classification tasks as they cannot properly handle sentences and documents that contains aspects with baring sentiment classes.

3 Approach

In this section, I define the problem tackled and introduce the proposed methods.

3.1 Problem Definition

Basically, the task this paper tackled is to predict ratings on multiple aspects given a review document. Dataset used here is BeerAdvocate, which consists of 37,500 comments on beers. The expected output of the problem consists of 5 discrete values from 1 to 5 with the interval being 0.5. The first four values on the output corresponds to ratings on Appearance, Aroma, Palate and Taste, and the last value corresponds to an Overall rating. Note here that Overall ratings are not a simple linear combination of the four aspect-based ratings, but are generated in a relatively independent way.

Based on my research on the dataset statistics, I found the the dataset has the following two features:

1. The scores are highly biased towards 4. The proportion of the score 4 among all the 37,500 scores on the aspects of appearance, aroma, palate, taste and overall are 42.82%, 34.75%, 38.94%, 33.01%, 36.98% respectively. Detailed distributions of the scores over the five aspects are given in Figure 1.

2. Input reviews are very long. The average length of reviews in the dataset is 131 words, the median length is 150 words and the longest review has 988 words. Additionally, the inner structure of each review is very complicated, as the comments on the five aspects are nested together in a single document.

Considering the above two features, as well as the fact that the scores are discrete values, I transform the problem from a value-prediction task to a classification task. Scores are mapped to classes in the following format: 1.0 → [Class 0], 1.5 → [Class 1], ..., 5.0 → [Class 8]. The reason behind it is that if the models perform value predictions and output real values, the outputs would tend to be invalid scores such as 3.8, which give low losses but do not have real-world meanings.

3.2 Baseline Method

For the baseline model, I adopt the architecture of a long short-term memory unit (LSTM) (Hochreiter and Schmidhuber, 1997) followed by a series of linear layers. Its architecture is illustrated in Figure 2, and the concrete explanations are addressed below:

1. For an input review $[w_1, w_2, ..., w_n]$ where w_i represents a single word, the model embeds it through a trainable embedding layer and get $[w_1^e, w_2^e, ..., w_n^e]$ where w_i^e represents the embedding of its corresponding word w_i.

2. Next, the model applies an one-directional LSTM layer to the embedded words, and retrieves the last hidden layer as the input's document-level representation h_d.

3. The model then applies a series of linear layers and activation functions to h_d to perform classification.

4. Finally the model applies a Softmax layer to the output of the last linear layer to get a probability distribution over score classes.

5. The score class with the highest distribution value is the final prediction.

3.3 Hierarchical Method

The main difference between the baseline model and the hierarchical model lies in the method how the document-level representations (paragraph representations) are generated. In the baseline model, inputs' document-level representations are generated from word embeddings directly. Instead of that, the hierarchical model encodes input reviews in a word-sentences-document order and builds the document-level representations on the top of

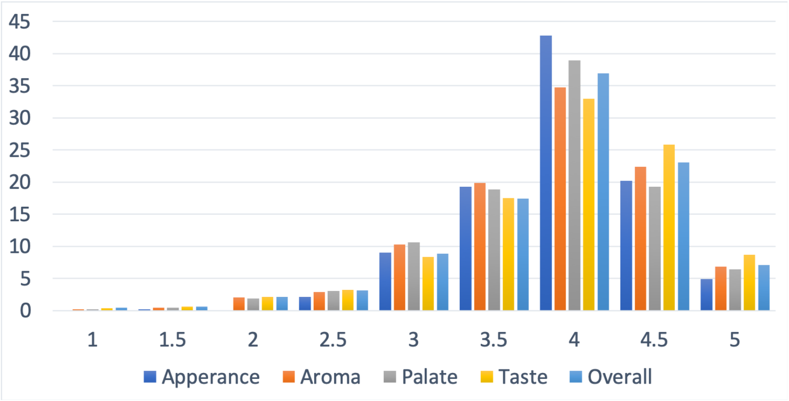


Figure 1: The distribution of scores over the five aspects. X-axis are the 9 discrete scores from 1 to 5, and the y-axis are the percentages (%) of scores.

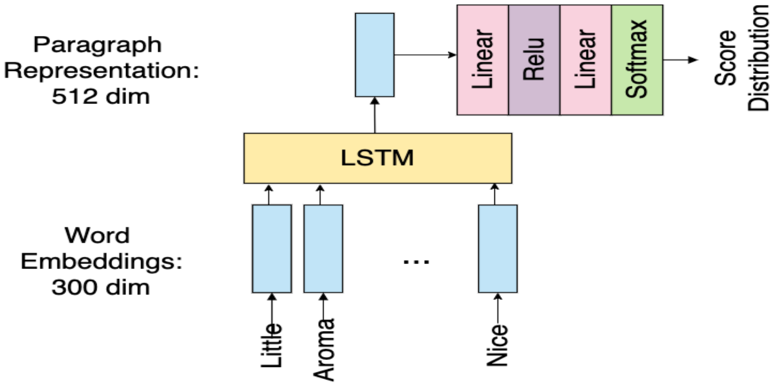


Figure 2: The architecture of the baseline model.

sentence-level representations. To achieve such encoding hierarchy, the hop mechanism is introduced into the model.

The model's architecture is illustrated in Figure 3, and the concrete explanations are addressed below:

1. Before sending an input review $[w_1, w_2, ..., w_n]$ into the encoding layer, the model chunks it into a sentence sequence $[s_1, s_2, ..., s_k]$ with a fixed hop size C_h. Each s_i corresponds to a word sequence $[w_{(i-1)C_h+1}, w_{(i-1)C_h+2}, ..., w_{iC_h}]$.

2. Then similar to the baseline method, the model embeds the sentence sequence $[s_1, s_2, ..., s_k]$ with an embedding layer and get $[s^e_1, s^e_2, ..., s^e_n]$ where s^e_i is the embedded sequence of its corresponding sentence sequence s_i.

3. Next, the model uses a bi-directional LSTM as the encoder to extract sentence-level information. The last hidden layers of the forward LSTM and the backward LSTM are concatenated as the input's sentence-level representation h_s.

4. h_s is sent into another bidirectional LSTM to generate the input's document-level representation h_d.

5. Finally, the model applies linear layers, activation functions and Softmax to h_d to produce the prediction.

4 Experiment

Note that the hierarchical model's accuracy and confusion matrix on the Overall aspect are different from what I presented on the poster session as the models had not fully converged by the time of the presentation.

4.1 Implementation Details

In the early data process stage, all reviews are padded to the same length. Denote the length of reviews as $\mathcal{L}_r$. $\mathcal{L}_r$ is ensured to be divisible by C_h in the hierarchical model.

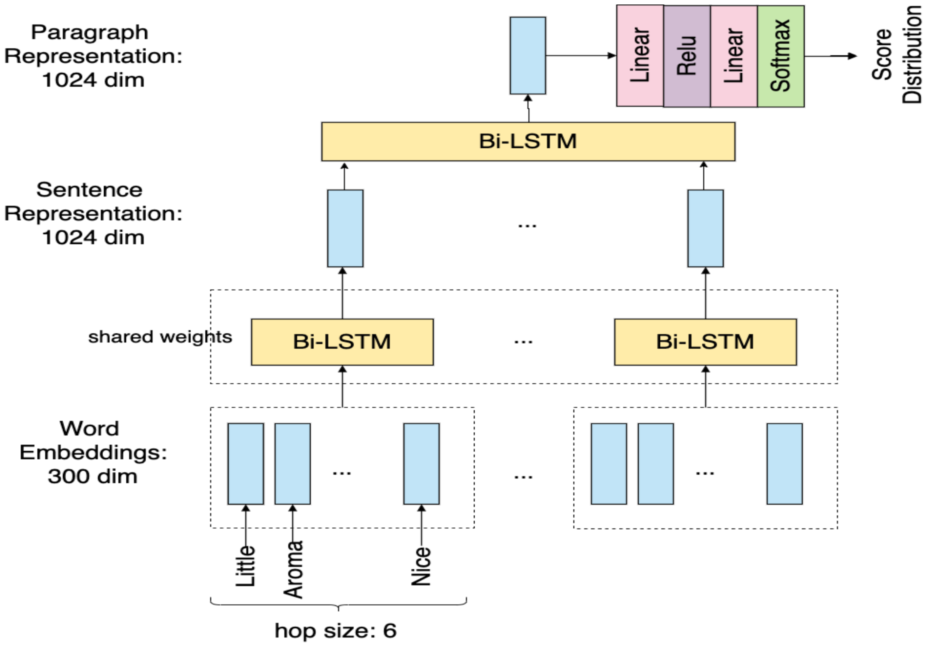


Figure 3: The architecture of the hierarchical model.

For the baseline mode, the embedding size is 300 and the hidden size of the LSTM is 512. Then the first linear layer has the output size of 256 and the last linear layer has the output size of 9 (the number of score classes).

For the hierarchical model, the hop size C_h used is 6, so the dimensions of inputs into the embedding layer become [$\mathcal{L}_r$ / C_h, C_h]. The embedding size is also 300, similar to the baseline mode. The two bi-LSTMs' hidden sizes are both 512, but since they go in two directions and the last hidden layers of the forward LSTM and the backward LSTM are concatenated, the dimensions of sentence representations and document representations are [$\mathcal{L}_r$ / C_h, 1024] and [1024] respectively. The output size of the two linear layers are 256 and 9, similar to the baseline model.

4.2 Accuracy

The accuracy comparison between the baseline model and the hierarchical model is shown in Figure 4. Basically, the baseline model achieves the accuracy of 0.4188, 0.3937, 0.4203, 0.3688 and 0.4 on the aspects of Appearance, Aroma, Palate, Taste and Overall respectively, and its average accuracy of the five aspects is 0.4003. The hierarchical model achieves the accuracy of 0.4219, 0.4359, 0.4437, 0.4625 and 0.4422 respectively with the average accuracy being 0.4412.

Comparing the accuracies of both models, it is obvious that the hierarchical model outperforms the baseline model in all aspects. Additionally, the improvements on the Aroma, Palate, Taste and Overall aspects are more significant than the improvement on the Appearance aspect.

4.3 Confusion Matrix

The confusion matrices of the two models on the five aspects are presented in Figure 5.

Comparing the confusion matrices of the two models, it is easy to observe that both models do not give good score distributions on the Appearance aspect. As illustrated on the first row of Figure 5, the baseline model and hierarchical model simply keep predicting score class 6 (which corresponds to score 4) on the Appearance aspect with any inputs given.

Except the Appearance aspect, the hierarchical model gives better score distributions on the other four aspects. To be more specific, the improvements on Palate, Taste and Overall aspects are more significant than the improvement on the Aroma aspect, as for the former three aspects the baseline model keeps 'cheating' and predicts score class 6 all the time while for the latter aspect it learns some reasonable distributions.

5 Discussion

5.1 Analysis of Baseline Model

From the confusion matrices of the baseline model, it is easy to observe that this model always gives

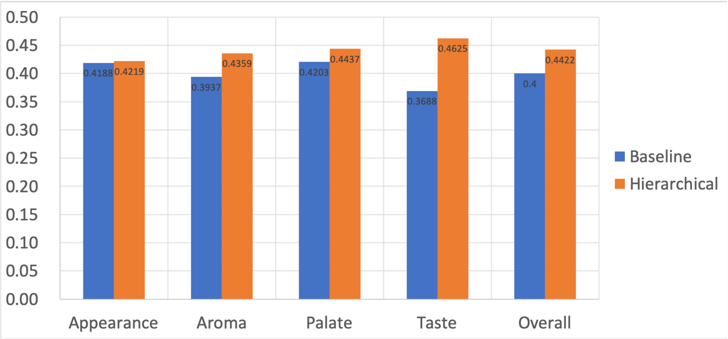


Figure 4: Accuracy comparison between the baseline model and the hierarchical model.

the prediction of score class 6 on the aspects of Appearance, Palate, Taste and Overall. There are several possible reasons behind this phenomenon:

1. Since the dataset is highly biased for the score of 4, the score class 6 (corresponds to score 4) is very likely to be a significant local optimal point in the gradient curve during the training process. In this case, the model would easily go towards the local mimum point on score class 6, which actually is not the global minimum.

2. Since the dataset is highly biased, the model might suffer from data insufficiency for minor options such as score class 0 (score 1), score class 1 (score 1.5) etc.

3. A single layer of one-directional LSTM is not capable of encoding information of the long input reviews and hence cannot provide enough information for the linear layers and activation layers to do the classification. The context behind this point is that RNN has the problem of gradient vanishing with long input sequences, and although LSTM model has been optimized over this issue, it is still a challenge for it to handle very long input sequences.

5.2 Analysis of Hierarchical Model

As observed in the above Experiment section, the hierarchical model gives significantly better performance on the aspects of Aroma, Palate, Taste and Overall. The improvements mainly come from the more effective encoding structure. By chunking inputs into short sequences and building document representations bottom-up mirroring documents' word-sentence-document hierarchy, we decrease the negative impact of gradient vanishing and enable the LSTMs to capture more useful information out of long input documents and hence providing more knowledge for the linear and activation layers to do the classification.

5.3 Comparison Among Aspects

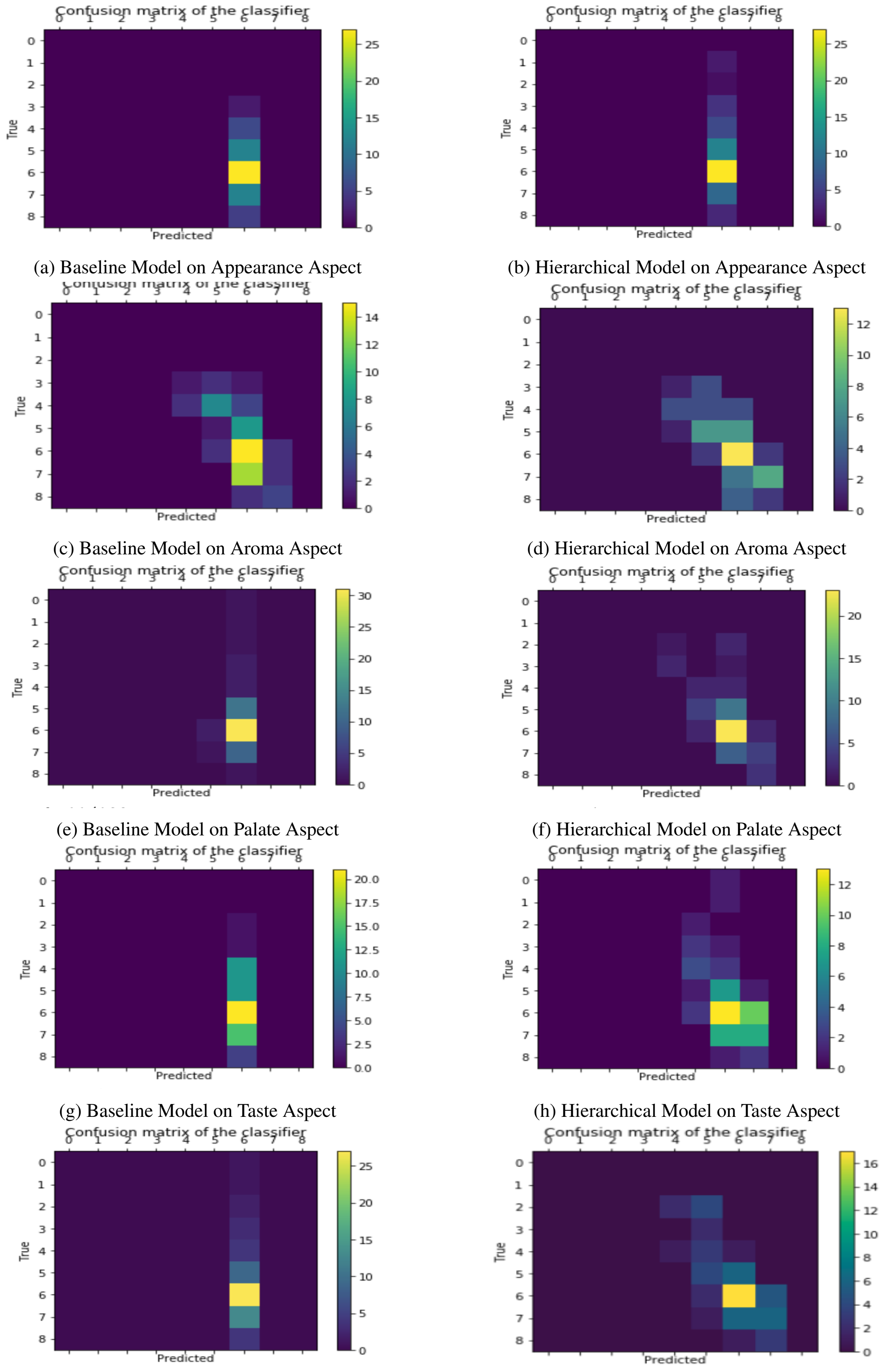
As mentioned above in the confusion matrix section, both the baseline model and hierarchical model give bad score distributions on the Appearance aspect. Additionally, as mentioned in the accuracy section, the improvement on the test accuracy of the Appearance aspect is least significant comparing to the other four aspects.

The possible reason behind these two observations might be the ambiguous and relatively neutral words used in this certain aspect. For example, the word ‘amber’ is widely used in the descriptions of appearance of beers (mentioned in 6632 reviews out of 37,500) while the word itself is rather neutral by its semantic meaning. In this case, the correlations between the input words and the preferred scores are lower in this aspect than in the other four aspects, which makes it more difficult for the LSTM to extract useful information and for the linear layer to do the correct classification.

6 Conclusion

In this paper, I introduce hierarchical models with hop mechanism to tackle the problem of document-level multi-aspects sentiment classification. The approach encodes semantics of input documents in a word-sentence-document hierarchy with bi-directional LSTMs. Experiments on the BeerAdvocate dataset show that the model significantly outperforms the baseline model in four out of five aspects, which suggests that the model provides a more semantically meaningful way to analyze long-text documents.

One possible future optimization can be conducted on introducing attention mechanism into



(a) Baseline Model on Appearance Aspect

(b) Hierarchical Model on Appearance Aspect

(c) Baseline Model on Aroma Aspect

(d) Hierarchical Model on Aroma Aspect

(e) Baseline Model on Palate Aspect

(f) Hierarchical Model on Palate Aspect

(g) Baseline Model on Taste Aspect

(h) Hierarchical Model on Taste Aspect

(i) Baseline Model on Overall Aspect

(j) Hierarchical Model on Overall Aspect

Figure 5: Confusion matrices of the baseline model and the hierarchical model on the aspects of Appearance, Aroma, Palate, Taste and Overall. The left figures are from the baseline model while the right figures are from the hierarchical model. Each row corresponds to an aspect.

the hierarchical model. By doing this, the model might be capable of differentiating more important and less important contents when constructing document representations (Yang et al., 2016). Another possible optimization direction is to use Tree-Structured LSTM (Tai et al., 2015) instead of the linear LSTM, as the natural language usually has Tree-like syntactic structures.

References

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Ainur Yessenalina, Yisong Yue, and Claire Cardie. 2010. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1046–1056. Association for Computational Linguistics.

Wei Zhang, Lifeng Jia, Clement Yu, and Weiyi Meng. 2008. Improve the effectiveness of the opinion retrieval and opinion polarity classification. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1415–1416. ACM.