Sentiment Classification with Gated CNN for Customer Reviews

Makoto Okada
Graduate School of Engineering
Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka, 599-8531, Japan
Email: okada@cs.osakafu-u.ac.jp

Hidekazu Yanagimoto
College of Sustainable System Sciences
Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka, 599-8531, Japan
Email: hidekazu@kis.osakafu-u.ac.jp

Kiyota Hashimoto
ESSAND
Prince of Songkla University
80 Moo 1 Vichitsongkram Road, Kathu, Phuket, 83120, Thailand
Email: kiyota.h@phuket.psu.ac.th

***Abstract*—Recurrent neural networks(RNNs) have been applied to sentiment classification but RNNs is usually heavier than convolutional neural networks (CNNs), turning more interest in the application of CNNs to language tasks. In this paper we propose a method to apply gated CNN (gCNN) with Maxpooling to sentiment classification of customer reviews. In our proposal, the application of gCNN is to sentiment classification, instead of constructing a language model. Our experiment is conducted with Amazon Product Review dataset and Japanese review dataset of TripAdvisor. The whole of each review is used as an input, instead of each sentence. The result is that a simple application of gCNN to sentiment classification achieved sufficient accuracies with the two datasets. Thus, an implication is that gCNN is proven to work fine for sentiment classification much faster than RNNs with fine results in the different language datasets.**

***Index Terms*—Sentiment analysis, Gated Convolutional Neural Network, Costumer review**

I. Introduction

As people post their opinions and reviews easier on the Internet, it is more important to estimate the polarity of them, whether they are positive or negative, more precisely. Sentiment classification, sentiment analysis, or opinion mining, has been pursued using a variety of techniques [13], [22], [23], [29]. Sentiment classification is roughly divided into two approaches: sentiment value summation and machine learning classification. Sentiment value summation is to sum up all the sentiment values of the sentiment words in a text. Thus a good sentiment dictionary, be it general or target-specific, is essential, and numerous studies on the construction of sentiment dictionary have been conducted [8], [18], [29]. Machine learning classification employs various machine learning methods for classification including Naïve Bayes and Support Vector Machines, and tf-idf, word sentiment value, etc. are used as features [11], [15], [19]–[21]. In both approaches, most of the previous studies employed a Bag-of-Words model, with which each sentence or passage is represented as a sparse vector of the dimension of the total number of words in the whole data, ignoring word order relations. Polarity inversion with a negative word and other modificational issues have also been investigated in heuristic or rule-based manner as well as using shallow syntactic parsing [13], [22]. However, so long as the input of machine learning approaches is a sparse vector under a Bag-of-Words model, such dependency issues remain, and more global dependency issues such as topic-subtopic relations are hard to cope with.

In recent years, deep learning approaches, employing deep neural networks, have been vigorously studied for natural language processing tasks including sentiment classification. One of the keys of employing deep learning approaches is recurrent neural networks (RNNs), with which sequential data, are processed as sequential [6]. Various natural language processing tasks have been investigated with RNN, from language modeling [16] to machine translation [1]. One of the RNNs typically employed for natural language processing is Long Short-Term Memory (LSTM) [10] and its successors including Gated Recurrent Units (GRU) [2]. LSTM and its successors incorporate the mechanism of probabilistically forgetting some dependencies, reflecting the characteristics of language that local dependency is usually more important than long dependency while some long dependencies should not be ignored. Sentiment classification has also been investigated with various RNNs [33], as well as recursive autoencoders [24] and recursive neural networks [25]. One of the issues of RNNs is difficulty in parallel computation with GPU, and the employment of convolutional neural networks (CNNs) to some natural language processing tasks have currently been pursued [3], [12].

For the employment of CNNs to natural language processing tasks, Dauphin et al. , following [3], proposed gated CNN (gCNN) for language modeling, in which a gated mechanism is incorporated into CNNs and it achieved almost the same result with LSTM at a much faster speed [4]. Sentiment analysis, which basically depends on occurrences of some sentiment words in certain contexts is expected to have a benefit of gCNN, but no previous investigation was made.

978-1-7281-0164-4/18/$31.00 ©2018 IEEE

In this paper, we apply gCNN [4] to sentiment classification. In our proposal, the application of gCNN is to sentiment classification, instead of constructing a language model. Our experiment is conducted with Amazon Product Review dataset [9], [14] and Japanese review dataset of TripAdvisor. The whole of each review is used as an input, instead of each sentence. The result is that a simple application of gCNN to sentiment classification achieved sufficient accuracies with the two datasets. Thus, an implication is that gCNN is proven to work fine for sentiment classification much faster than RNNs with fine results in the different language datasets.

This paper is organized as follows: the next section briefly surveys some related works. Section 3 introduce our proposed methods, gCNN in our framework. Section 4 describes our evaluative experiments and their results and discussion. Section 5 concludes the paper.

II. Related Works

Deep learning has been investigated for sentiment estimation [33]. In most cases, RNNs or recursive neural networks were employed.

First, Socher et al. proposed semi-supervised recursive autoencoders for sentiment analysis [24]. They applied an autoencoder to phrases recursively to construct the semantic vector representation of each phrase. Sentiment distribution is predicted by a softmax classifier. Socher et al. then proposed recursive neural network in matrix-vector space [25]. In the method a composition function of two words is $\mathbf{A}\mathbf{w}_1 + \mathbf{B}\mathbf{w}_2$ for words $\mathbf{w}_1, \mathbf{w}_2$. $\mathbf{A}$ and $\mathbf{B}$ are matrices to be learnt with the training data, and sentiment distribution is estimated by a softmax classifier. In the output layer of the method sentimental distribution is predicted with a softmax function. Socher et al. also proposed recursive neural tensor network [26]. The method uses tensor-based composition function for all nodes in the parse tree. The function is applied to each node recursively and the semantic vector representation of a sentence is generated. Sentiment distribution is also predicted by a softmax classifier.

Tang et al. applied gated recurrent neural network at a document level [28]. Their approach is first from word representations to construct sentence representations with CNN and LSTM, then to construct document representations with gated RNN to be used for sentiment analysis with a softmax classifier. Xu et al. applied cached long short-term memory neural network to document-level sentiment classification. Their LSTM is enhanced with a cache mechanism by preparing different forgetting rates for different groups of memory [31]. Dou et al. applied LSTM with memory network [30] with a recurrent attention model [27] to document-level sentiment classification [5].

An important common characteristics of these papers is to first embed a word into a distributed word representation with 200 or some dimensions. The basic concept of distributed word representation is that the (at least partial) semantics of a word is decided by its context [7], and recently practical computational techniques have been proposed [17]. Many researchers have used such distributed word representations as a pre-training set, but they are also directly constructed at earlier layers in a neural network, which is adopted in the papers above. Note that this direct construction of distributed word representation is the result of the learning of the whole neural network model, and thus there is no guarantee that each word's distributed representation vector reflects a fragment of the meaning that we may understand in a general sense, which is contrary to word2vec [17] or other distributed word representation model whose aim is to construct distributed word representations that embed a word meaning in a sense. Another important common characteristics of most of these papers is to probabilistically control which feature be kept and forgotten. Thus if CNN can hold these two characteristics, it is expected to obtain a faster sentiment classifier, which is the initial idea of this paper.

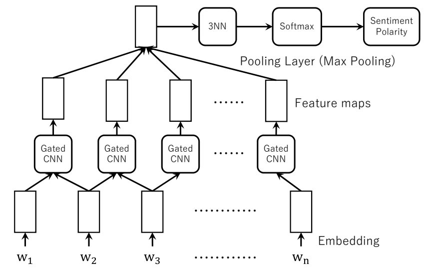

Fig. 1. Architecture of Sentiment Analysis with Gated convolutional neural networks

III. Sentiment Analysis with Gated convolutional neural networks

We first propose a sentiment analysis method with gated convolutional neural networks [4]. Dauphin et al. employed gCNN to construct a language model to predict the next word from the previous words. Thus, their gCNN was designed to produce another sequence. On the other hand, our gCNN aims to produce a sentiment value, positive or negative for each data entry. Thus the first difference between their model and our model is what the output is, and thus the pooling layer is designed to finally return one value, positive or negative. For this pooling layer, our proposal is to employ Max Pooling. The overview of our architecture is shown in Fig. 1.

A. Word embedding

The initial input is the sequence of words in a text. Each word is represented as a one-hot vector of V-th dimension, where V is the total number of the different words in the dataset, or the size of vocabulary in the dataset. Before going into convolutional and pooling layers, each one-hot vector is first converted into a vector of m-th dimension, where m, manually specified, is far smaller than N. WIth a usual backpropagation method, the error is returned to each embedded vector for an update, whose values are stored at the

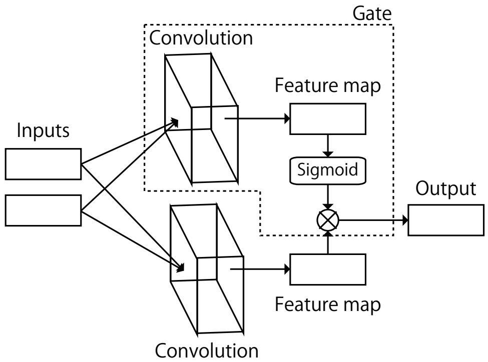

Fig. 2. Architecture of gated convolutional neural networks

look-up table. For each input, the updated embedded vector of each word by the previous inputs is called from the look-up table. By the end of training, each word in the look-up table is represented as a distributed word representation vector.

Technically, when $\mathbf{w}_i$ is a 1-of-V coding vector, the embedding process is shown below. First, we construct a vector from a word with an embedding matrix, $\mathbf{D} \in \mathbb{R}^{m \times V}$.

$$\mathbf{e}_i = \mathbf{D}\mathbf{w}_i \tag{1}$$

Because a sentence is a sequence of words, the sentence is represented as a sequence of embedding vectors. In the paper, we denote the sequence of vectors as $\mathbf{X}$. So the $\mathbf{X}$ is the input of our gCNN.

$$(\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_V) \Rightarrow (\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_V) = \mathbf{X} \tag{2}$$

B. gCNN for language sequences

The core idea of gCNN is to adopt the gate mechanism of LSTM. To keep long dependency minimizing the risk of more important shorter dependency being excessively unweighted, LSTM has three gates to control information flow: forget gate, input gate, and output gate. gCNN, as shown in Fig. 2, consists of a stack of usual convolutional layers and a gated linear unit (GLU).

The output of gCNN is the vector representation of the whole data entry, a review in this paper. This vector representation is input to SPP to output a binary value of sentiment classification.

As a usual CNN accepts a matrix as its input, gCNN accepts a matrix consisting of multiple vectors. The input is transformed into two different vectors with two different kernels (tensors). One vector reflects the distributed representation of the input data while the other vector information to control GLU in gCNN.

When gCNN is applied to a text, words are integrated with an additive. Consider two words, $\mathbf{e}_1$ and $\mathbf{e}_2$, are integrated with gCNN.

$$\mathbf{m} \otimes \left(\mathbf{k}_1^{\mathrm{T}}\mathbf{e}_1 + \mathbf{k}_2^{\mathrm{T}}\mathbf{e}_2\right) \tag{3}$$

$\mathbf{m}$ is calculated with $\sigma\left(\mathbf{X} * \mathbf{F}_g\right)$ in Eq. 4 and $\mathbf{k}_1$ and $\mathbf{k}_2$ are weights and are defined based on $\mathbf{F}_c$ in Eq. 4.

Then GLU, a gating mechanism, work as follows:

$$\mathbf{X}' = \sigma\left(\mathbf{X} * \mathbf{F}_g\right) \otimes \left(\mathbf{X} * \mathbf{F}_c\right) \tag{4}$$

where $\mathbf{X} \in \mathbb{R}^{N \times m}$ is the input of gCNN, $\mathbf{F}_g \in \mathbb{R}^{k \times m \times n}$ and $\mathbf{F}_c \in \mathbb{R}^{k \times m \times n}$ are kernels for the convolutional operation, $\sigma(\cdot)$ is the sigmoid function, $*$ denotes the convolution, and $\otimes$ is the element-wise product of vectors. The output is $\mathbf{X}' \in \mathbb{R}^{k \times (N-n+1)}$, which is shorter than the length of the input $\mathbf{X}$, when padding is not used. In this case, $\mathbf{F}_g$ is a mapping to control GLU and $\mathbf{F}_c$ is a mapping to compose inputs, which are the target of training. Because the output of the sigmoid function is restricted from 0 to 1, the outputs control how strongly each fragment of the input affects the final output.

After processing $\mathbf{X}$ with gCNN, a shorter sequence of vectors is obtained, which is then to be used as the input for classification, instead of a language model whose output is also a sequence of text. Thus, before going into the final classification task with a usual three-layer neural network, we need the pooling layer after gCNN, for which we apply Max Pooling. The pooling layer is a simple Maxpooling layer with which the maximum value among the convolutional output is selected for its output.

C. Final sentiment classification

With the output of Max Pooling, the final sentiment classification is conducted with a simple three-layer neural network with the softmax function.

$$\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{f}) \tag{5}$$

$$\mathbf{o} = \mathrm{Softmax}\left(\sigma(\mathbf{W}_2\mathbf{h})\right) \tag{6}$$

$\mathbf{W}_1$ and $\mathbf{W}_2$ are weights of the three-layer neural network. $\mathbf{W}_1$ is between the input layer and the hidden layer. $\mathbf{W}_2$ is between the hidden layer and the output layer.

For training, we employ cross entropy as a loss function between the prediction and the supervised sentiment polarity label $\mathbf{t}$.

$$\mathbf{t} = \begin{cases} (1,0) & \text{(negative)} \\ (0,1) & \text{(positive)} \end{cases} \tag{7}$$

$$E(\mathbf{o}, \mathbf{t}) = t_1 \log o_1 + t_2 \log o_2 \tag{8}$$

The parameters to be trained in our proposal is thus defined as $\theta = (\mathbf{D}; \mathbf{F}_g; \mathbf{F}_c; \mathbf{W}_1, \mathbf{W}_2)$, and training is conducted with the backpropagation algorithm.

$$\theta \leftarrow \theta - \epsilon \frac{\partial E(\mathbf{o}.\mathbf{t})}{\partial \theta} \tag{9}$$

IV. Experiments

In order to evaluate effectiveness of our proposal method and the degree of the effect of the different language, we classified two review datasets and compared the results. One review dataset was English reviews of Amazon.com. Another set was Japanese review dataset extracted from Trip Advisor.

TABLE I
CONTENTS OF REVIEW DATASET AMAZON PRODUCT REVIEW

Reviews (words)	Whole	Training set	Test set
Positively labeled reviews	5,000	4,500	500
Negatively labeled reviews	5,000	4,500	500
Total	10,000	9,000	1,000
Average words in reviews	59.1 words		

TABLE II
CONTENTS OF REVIEW DATASET TRIPADVISOR REVIEW

Reviews (words)	Whole	Training set	Test set
Positively labeled reviews	4,545	4,090	455
Negatively labeled reviews	4,543	4,089	454
Total	9,088	8,179	909

A. Datasets

We used parts of Amazon Product Review datasets by Julian McAuley [9], [14]. The whole dataset consists of product reviews and metadata from Amazon, including 142.8 million reviews during May 1996 - July 2014. It contains reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs).

We used parts of travelars' reviews of TripAdvisor. TripAdvisor is one of famous tourism web site and it contains very large amount of reviews that was written by various languages. In this experiment, we selected reviews written by Japanese.

For the experiment, we used parts of small subset dataset of the Amazon Product Review dataset. These small subsets are divided into 24 categories. Among them, we selected four categories: Sports and Outdoors, Health and Personal Care, Video Games, and CD & Vinyl. From each subset, 10,000 reviews were randomly sampled in a way that it contains 5,000 positive reviews and 5,000 negative reviews. In this sampling, we regard only the 5-star reviews as positive and 1-star reviews as negative, and 2, 3, and 4-star reviews are not included. We separated each experimental data into 9,000 reviews as the training data and 1,000 reviews as the test data. The length of each review is less than or equal to 1,000 characters.

Table I shows the content of our dataset.

We also used parts of dataset of TripAdvisor Japanese reviews. From the dataset, 9,090 reviews were randomly sampled in a way that it contains 4,545 positive reviews and 4,543 negative reviews. In this sampling, we regard only the 4 and 5-star reviews as positive and 1 and 2-star reviews as negative, and 3-star reviews are not included. We separated each experimental data into 8,179 reviews as the training data and 909 reviews as the test data.

Table II shows the content of our dataset.

Each text of the Japanese review data were processed Japanese Part of Speech tagger "MeCab"[1] and separated from sentences to words. We had prepared two datasets. One dataset contained every words as base form such as "go" and "come", another dataset contained every words as original form in the review such as "goes" and "came". The two datasets were same size.

[1]http://taku910.github.io/mecab/

TABLE III
PARAMETER SETTINGS FOR THE PROPOSED METHOD

Model parameters	Setting
Threshold frequency	3
Embedding size(m)	200
Kernel size($\mathbf{F}_g$, $\mathbf{F}_c$)	$200 \times 200 \times 2$
Minibatch size	500
Classifier	200-100-2
Optimization	ADAM

TABLE IV
EXPERIMENTAL RESULTS USING AMAZON PRODUCT REVIEW DATASETS

Category	gCNN		
	Precision	Recall	F1
Sports and Outdoors	0.99	0.99	0.99
Health and Personal Care	0.87	0.86	0.86
Video Game	0.83	0.82	0.82
CD & Vinyl	0.83	0.83	0.83

B. Implementation

Our experimental system is implemented with Chainer[2] on Python 3.5. Parameters were set as in Table III.

The threshold of minimum occurrence frequency is determined as 3. It means that words appearing less than three times are regarded as unknown words. The embedding size affects the performance of prediction strongly, and we set 200 dimensions in the experiments. Thus the kernel size is $200 \times 200 \times 2$, and only the relations between two words are considered. In the proposed system we use a simple 3-layer neural network and architecture is fixed. The design of a classifier is important, but the discussion is one of future works.

C. Result and Discussion

1) The experimental results of Amazon Product Review: As shown in IV, our gCNN model achieved high accuracy as experimental results, which shows the effectiveness of our proposed methods. In this results, "Sports and Outdoors" was the highest result and others were a little bit lower than them. The reason of the difference of the accuracy was expected the vocabulary of each subsets. When the vocabulary was clearly different with the positive data and the negative data, it is easy for the classifier to separate these data into two classes. However, if the vocabulary of the datasets were overlapped, it will make a confusion to the classifier. The detail research of the difference of the vocabulary is our future work.

2) Experimental results of TripAdvisor review: The second experiment compared the difference of the languages of the review data.

As shown in V, the experimental result of base form were almost same as the result of original form. In this experiments, the difference of the form of the words did not affect to the

[2]https://chainer.org

TABLE V
EXPERIMENTAL RESULTS USING TRIPADVISOR REVIEW

	Precision	Recall	F1
Original form	0.87	0.86	0.86
Base form	0.87	0.87	0.87

accuracy. We will research the effect when number of data are changed stepwise.

To compare the experimental results of Amazon Product Review, both of the results were sufficient for using, and the results of TripAdvisor reviews were higher for the most part of the results of Amazon Product Review. It shows that our proposal model is effective for the Japanese data and it will be effective to other language data.

V. CONCLUSION

In this paper we proposed sentiment classification with gCNN and it applied to the two datasets: Amazon Product Review and TripAdvisor reviews. The result tells that gCNN performs enough to use. This suggests that gCNN for language processing was originally designed for sequential language model constructions, it also works fine with document classification tasks. This study is still in its infancy and more investigation is needed, but the effectiveness of gCNN for document classification tasks is confirmed for more applications.

REFERENCES

[1] M. Auli, M. Gallery, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," in *Proc. EMNLP '13*, Oct. 2013, pp.1044-1054.

[2] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," in *Proc. EMNLP '14*, Oct. 2014, pp. 1724-1734.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach Learn Res*, vol. 12, pp.2493-2537, 2011.

[4] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional neural networks," arXiv:1612:08083, 2016.

[5] Z.-Y. Dou, "Capturing user and product information for document level sentiment analysis with deep memory network," in *Proc. EMNLP 2017*, Sept. 2017, pp.521-526.

[6] J. L. Elman, "Finding structure in time," *Cog. Sci.*, vol. 14, no. 2, pp.179-211, 1990.

[7] Z. S. Harris, "distributed structure," *Word*, vol. 10, no. 2-3, pp.146-162, 1954.

[8] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in *Proc. ACL '98*, July 1997, pp.174-181.

[9] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with oneclass collaborative filtering," in *Proc. WWW 2016*, Apr. 2016, pp.507-517.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput*, vol. 9, no.8, pp.1735-1780, 1997.

[11] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in *Proc. ECML-98*, Apr. 1998, pp.137-142.

[12] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. EMNLP '14*, Oct. 2014, pp.1746-1751.

[13] B. Liu, *Sentiment analaysis and opinion mining*, VT: Morgan & Claypool, 2012.

[14] J. McAuley, C. Targett, J. Shi, and A. van den Hengel, "Image-based recommendations on styles and substitutes," in *Proc. SIGIR '15*, Aug. 2015, pp.43-52.

[15] A. McCallum and K. Nigam, "A comparison of event models for Naive Bayes text classification," *Proc. AAAI-98 Workshop on Learning for Text Categorization*, July 1998, pp.41-48.

[16] T. Mikolov, M. Karafiát, L. Burget, J. Cemockỳ, and S. Khudanpur, "Recurrent neural network based language model," *Proc. Interspeech 2010*, vol. 2, pp.1045-1048, 2010.

[17] T. Mikolov, K. Chen, G. S. Corrado, and F. Dean, "Efficient estimation of word representations in vector space," arXive:1301.3781v3, 2013.

[18] S. J. Pan, X. Ni, J. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proc. WWW '10*, Apr. 2010, pp.751-760.

[19] B. Pang, L. Lee, and S. Vaithyanathan "Thumbs up? Sentiment classification using machine learning techniques," in *Proc. EMNLP '02*, July 2002, pp.76-86.

[20] B. Pang, and L. Lee, "Sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. ACL '04*, July, 2004, 8 pages.

[21] B. Pang, L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," *Proc. ACL '05*, July 2005, 10 pages.

[22] B. Pang, and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no.1-2, pp.1-135, 2008.

[23] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma, "Sentiment analysis: A review and comparative analysis of Web services," *Inf Sci*, vol. 311, pp.18-38, 2015.

[24] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, *Semi-supervised recursive autoencoders for predicting sentiment distribution*, in *Proc. EMNLP '11*, July 2011, pp. 151-161.

[25] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionally through recursive matrix-vector spaces," in *Proc. EMNLP '12*, July 2012, pp.1201-1211.

[26] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP '13*, Oct. 2013, pp.1631-1642.

[27] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. NIPS '15*, Dec. 2015, pp.2440-2448.

[28] D. Tang, B. Qin, and T.— Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proc. EMNLP '15*, Sept. 2015, pp.1422-1432.

[29] P. D. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," in *Proc. ACL '02*, July 2002, pp.417-424.

[30] J. Weston, S. Chopra, and A. Bordes, "Memory networks," arXive: 1401.3916v11.

[31] J. Xu, D. Chen, X. Qiu, and X. Huang, "Cached long short-term memory neural networks for document-level sentiment classification," in *Proc. EMNLP 2016*, Nov. 2016, pp.1660-1669.

[32] M. Zhang, Y. Zhang, and D. T. Vo, "Gated neural networks for targeted sentiment analysis," in *Proc. AAAI '16*, Feb. 2016, pp.3087-3093.

[33] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," arXiv: 1801.07883, 2017.