Sentiment Analysis of Peer Review Texts for Scholarly Papers

Ke Wang and Xiaojun Wan
Institute of Computer Science and Technology, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
Beijing, China
wangke17@pku.edu.cn,wanxiaojun@pku.edu.cn

ABSTRACT

Sentiment analysis has been widely explored in many text domains, including product reviews, movie reviews, tweets, and so on. However, there are very few studies trying to perform sentiment analysis in the domain of peer reviews for scholarly papers, which are usually long and introducing both pros and cons of a paper submission. In this paper, we for the first time investigate the task of automatically predicting the overall recommendation/decision (accept, reject, or sometimes borderline) and further identifying the sentences with positive and negative sentiment polarities from a peer review text written by a reviewer for a paper submission. We propose a multiple instance learning network with a novel abstract-based memory mechanism (MILAM) to address this challenging task. Two evaluation datasets are constructed from the ICLR open reviews and evaluation results verified the efficacy of our proposed model. Our model much outperforms a few existing models in different experimental settings. We also find the generally good consistency between the review texts and the recommended decisions, except for the borderline reviews.

CCS CONCEPTS

• **Information systems** → **Sentiment analysis**; **Clustering and classification**;

KEYWORDS

Sentiment analysis; peer review mining; multiple instance learning; abstract-based memory mechanism

ACM Reference Format:
Ke Wang and Xiaojun Wan. 2018. Sentiment Analysis of Peer Review Texts for Scholarly Papers. In *SIGIR '18: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, July 8–12, 2018, Ann Arbor, MI, USA,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3209978.3210056

1 INTRODUCTION

Nowadays, with the rapid development of scientific research, more and more scholarly papers have emerged. At the same time, one of the best way to evaluate those papers is peer review (also known as refereeing) [3]. For most journals and conferences in the computer science field, peer review is used to help decide whether a paper submission should be accepted or rejected. Usually, a professional reviewer writes a detailed review text to introduce both the pros and the cons of a submitted paper and then gives an overall score to recommend a decision status (typically accept or reject).

However, it is still unknown whether the review texts and the recommendation scores are consistent with each other or not. If not, the review submission system should warn the reviewer if there is any potential mistake in review scores. Moreover, if we can automatically highlight the reviewer's controversies (e.g., the major pros and cons in the review text) on the submitted paper, it will not only help the chair to write a comprehensive meta-review, but also be convenient for authors to further improve their paper.

In order to address the above problems, we investigate the sentiment analysis task in the new domain of peer review texts for scholarly papers, which makes an important step in the area of sentiment analysis and artificial intelligence. As a first step in this research direction, the task investigated in this paper is simply defined as predicting the overall recommendation status (accept, reject, or sometimes borderline) and identifying the sentences with

This paper presents low-rank bilinear pooling that uses Hadamard product. The paper implements · · ·   · · · · · ·   I like the insights about low-rank bilinear pooling using Hadamard product presented in the paper. *However, it could not be justified that low-rank bilinear pooling leads to better performance than compact bilinear pooling. It does lead to reduction in number of parameters but it is justification of why low-rank bilinear pooling is better than other forms of pooling.*
Prediction: Accept   Summary: [+0.19] I like the insights about low-rank bilinear pooling leads· · ·   [+0.12] The paper presents new insights into element-wise · · ·   [+0.06] The paper presents a new model for the task of VQA · · ·   [+· · ·] · · ·   [- 0.12] it could not be experimentally verified that low-rank · · ·   [- 0.11] I would like the authors to provide experimental · · ·   [- 0.05] It is not very clear from reading the paper.   [-· · ·] · · ·

Table 1: An example of peer review text and the analysis results. The above block is an example peer review for a scholarly paper, and the below block is the analysis results we hope for: predicting the overall recommendation status and identifying the opinionated sentences in the review (plus the sentiment polarity score of each sentence).

SIGIR '18, July 8–12, 2018, Ann Arbor, MI, USA

ACM ISBN 978-1-4503-5657-2/18/07. . . $15.00
https://doi.org/10.1145/3209978.3210056

positive and negative sentiment polarities from the peer review text. For illustration, the above block in Table 1 is an example review (The original text of this review text contains 399 words, due to space limit, we have to intercept a small fragment here as an example). In the review text, the contribution of the paper is first summarized, and then the reviewer expresses his opinions about the paper, including both positive aspects (see the underlined text) and negative aspects (see the italic text), We hope to be able to automatically determine the overall recommendation status (i.e., Accept) of the reviewer and discovering reviewer's opinions from this review text (see the below block in Table 1).

Sentiment analysis is a hot topic in the researches of natural language processing and has been investigated for a long time. Some researchers have applied sentiment classification to a wide range of text domains such as product reviews [4, 9, 20, 24, 39, 41, 51], movie reviews [20, 27, 33, 41], tweets [10, 14, 18, 32], news articles [15], etc., and they have achieved some significant advances in these domains. But there is very few research on sentiment classification of the peer review texts, not only because the peer review corpus is hard to obtain but also because of the following main challenges.

- **Long length.** In our evaluation datasets constructed from the ICLR open reviews, the average length of the review text is about 299 words, and longer text makes it more difficult to capture the overall sentiment polarity.
- **Mixture of non-opinionated and opinionated texts.** A review text contains a summary of the contributions or content in the paper. In many reviews, the points regarding the writing are also contained, e.g., typos and grammatical errors. The reviewer's opinionated text is mixed with the non-opinionated text, and it is not easy to separate them.
- **Mixture of pros and cons.** In a review, the reviewer usually talks about both merits and shortcomings of the paper. A paper usually has a few merits and a few shortcomings, so it is difficult to capture the key points of the reviewer as evidence for overall recommendation.

In this study, we propose a **M**ultiple **I**nstance **L**earning Network with a novel **A**bstract-based **M**emory mechanism (**MILAM**) to address this challenging task. Our neural network model is elaborately designed based on multiple instance learning network (MIL) [2]. Moreover, by considering the abstract information of a scholarly paper as memories, it leverages the relevant memories to control the representation of each sentence in the review text. We collect all peer review texts of ICLR-2017[1], ICLR-2018[2] (International Conference on Learning Representations) as evaluation datasets which are publicly accessible on the OpenReview website. Evaluation results show that our proposed neural network model outperforms several baseline methods in different experimental settings, including some of the state-of-art neural network models for text classification and sentiment classification. The predicted recommendations for all reviews of a paper can be aggregated to predict the final decision (accept or reject) of the paper. Moreover, our model can well identify opinionated sentences with polarity scores from the review text and the reviewer's opinions can be used to help authors to further improve their paper.

[1]https://openreview.net/group?id=ICLR.cc/2017/conference
[2]https://openreview.net/group?id=ICLR.cc/2018/Conference

We also find that the accuracy of overall recommendation prediction (accept and reject) achieved by our model can be improved by about 10 points if the borderline reviews are removed. Empirical analysis shows that the borderline reviews are hard to be classified. The results indicate the generally good consistency between the review texts and the recommended decisions, except for the borderline reviews.

The major contributions of this paper are summarized as follows:

1) To the best of our knowledge, we are the first to investigate the sentiment analysis task in the domain of peer review texts for scholarly papers. Moreover, we built two evaluation datasets by using all peer reviews of ICLR-2017 and ICLR-2018.
2) We propose a multiple instance learning network with a novel abstract-based memory mechanism (MILAM) to address this challenging task.
3) Extensive experiments are conducted and evaluation results demonstrate the efficacy of our proposed model and show the great helpfulness of using abstract as memory. We also draw some interesting conclusions by empirical analysis.

In the rest of this paper, we will first introduce related work, and then describe our proposed model and discuss the evaluation results. After that, we conclude this paper.

2 RELATED WORK

2.1 Sentiment Classification

Most of the sentiment analysis researches focus on sentiment classification, which aims to determine the sentiment polarity (i.e., positive or negative) of a text. In the past fifteen years, sentiment classification techniques have been applied in various text domains, including product reviews [4, 9, 20, 24, 39, 41, 51], movie reviews [20, 27, 33, 41], tweets [10, 14, 18, 32], news articles [15], and so on. The methods for sentiment classification can be coarsely categorized into lexicon based methods [36, 41] or machine learning based methods [4, 14, 18, 27], and machine learning based methods, especially the neural network models [10, 20, 32, 33, 39, 51], have demonstrate superior performance. However, to the best of our knowledge, sentiment classification techniques have not been applied in the domain of peer reviews for scholarly papers.

2.2 Multiple Instance Learning

In order to extract the reviewer's opinions (the opinionated sentences in the review text), our models adopt a multiple instance learning (MIL) framework [19]. MIL deals with problems where labels are associated with groups of instances or *bags* (reviews in our case), while instance labels (sentence-level polarities) are unobserved. The initial MIL made the strong assumption that a bag is negative only if all of its instances are negative, and positive otherwise ([11, 25, 50]), and subsequent work relaxed this assumption and made it more suitable for the task at hand. Recently, MIL techniques have been widely used in various tasks, such as drug activity prediction [11], image retrieval [25], object detection [8, 49], text classification [1], image captioning [44], paraphrase detection [47], information extraction [17] and sentiment analysis[2, 22, 28, 37].

But none of their work was applied to the challenging task of sentiment analysis of peer review texts, while we proposed a novel abstract-based memory mechanism to address the task.

2.3 Memory Network

Traditional deep learning models (RNN, LSTM, GRU, etc.) use hidden states or attention mechanisms to leverage extra data(memories), but those methods have too little memory space to accurately record the entire contents of data, that is, they lose a lot of information when encoding input into dense vectors. However, memory network uses a external memory module to save extra data (the abstract text in our case) and jointly learns with the goal of using it for prediction. Memory network is a general machine learning framework introduced by [43] and has made great success in question answering [23, 26, 30, 35, 43], dialogue system [5, 12, 42] and so on[38, 40]. Different from the above, here we propose a novel abstract-based memory mechanism, with the intuition to leverage the abstract text to reduce the effects of irrelevant noise (like repeating the contents of the paper) in the review text, which may confuse the classifier.

2.4 Study on Peer Reviews

Several studies have tried to automatically predict peer reviews' helpfulness or assess the quality of peer reviews. In these studies, the peer reviews are not limited to peer reviews for scholarly papers, but contain peer reviews for students' work. For example, [6] presented machine learning technologies applied for classifying peer comments in writing (specific vs. non-specific, praise vs. criticism), and SVM achieved a noteworthy performance. [46] proposed a system for generating automatic assessments of reviewing performance with respect to problem localization at the reviewer-level, and demonstrated the feasibility of detecting reviewers who have low problem localization in reviewing. [45] further showed that the utility of generic features in predicting review helpfulness varies between different review types. [31] used effective data preprocessing techniques along with latent semantic analysis and cosine similarity to determine the quality, tone and quantity of review comments. [48] proposed to use decision-tree based classifier to evaluate a review's quality. However, these tasks are related but different from the sentiment analysis task addressed in this study.

3 FRAMEWORK

In order to address the task of sentiment analysis of peer review texts, we proposed a multiple instance learning network with a novel abstract-based memory mechanism, named **MILAM**. Our neural network model is elaborately designed based on multiple instance learning network (MIL), and it leverages the abstract information to control the representation of each sentence in the review text by introducing memory mechanism. We first describe the general architecture of our model and then introduce the abstract-based memory mechanism used in it.

3.1 Architecture

The architecture of our proposed model (MILAM) is shown in Figure 1. The whole model can be divided into three layers. The input representation layer aims to transform each sentence of the review text and the paper's abstract text into their distributed representations using sentence embeddings with Convolutional Neural Network(CNN) [7, 20]. In sentence classification layer, by considering representations of the abstract information as memories, we obtain new high-level representations of each sentence in the review text by leveraging relevant memories. After that, we use the softmax function to get the sentence-level distribution over sentiment labels. At the review classification layer, we predict the review's overall sentiment label (i.e. overall recommendation / decision) based on the sentiment label distributions of sentences and a document attention mechanism.

Input Representation Layer: The inputs of our model are the word sequences of the review text T_{review} and the paper's abstract text $T_{abstract}$. The texts T_{review} and $T_{abstract}$ contain n and m sentences respectively, and each sentence is composed of several words. We represent each word $\boldsymbol{w}_i \in \mathbb{R}^d$ as a fixed-size vector by looking up from pre-trained word embeddings, where d is the dimension of the word vector. And a sentence $\boldsymbol{S}$ of length L (padded where necessary) is represented as

$$\boldsymbol{S} = \boldsymbol{w}_1 \oplus \boldsymbol{w}_2 \oplus \cdots \oplus \boldsymbol{w}_L, \boldsymbol{S} \in \mathbb{R}^{L \times d}, \tag{1}$$

where $\oplus$ is the concatenation operator. Thus we use $\{\boldsymbol{S}_i^r\}_{i=1}^n$ and $\{\boldsymbol{S}_j^a\}_{j=1}^m$ to represent the review text and the paper's abstract text respectively.

After converting sentences into embedding vectors, the convolutional layer aims to get high-level representations of the sentences in the review text and the abstract text. The convolutional layer extracts local features by sliding a window of length l (i.e., 3 in this study) over the text and performs a convolution within each sliding window, and the output of the k-th sliding window is computed as

$$f_k = tanh(\boldsymbol{W}_c \cdot \boldsymbol{W}_{k-l+1:k} + \boldsymbol{b}_c), \tag{2}$$

where $\boldsymbol{W}_{k-l+1:k}$ denotes the concatenation of l word embeddings within the k-th window in sentence S, $\boldsymbol{W}_c$ is the convolution matrix and $\boldsymbol{b}_c$ is the bias. We use multiple filters, and for the q-th filter, it is applied to each possible window of words in the review text $\{\boldsymbol{W}_{1:l}, \boldsymbol{W}_{2:l+1}, \cdots, \boldsymbol{W}_{L-l+1:L}\}$ to produce a feature map

$$\boldsymbol{f}^{(q)} = [f_1^{(q)}, f_2^{(q)}, \cdots, f_{L-l+1}^{(q)}], \tag{3}$$

with $f^{(q)} \in \mathbb{R}^{L-l+1}$. After that, we apply a max-pooling operation to obtain the most salient feature for the input review text

$$u_q = max\{\boldsymbol{f}^{(q)}\}. \tag{4}$$

Supposing the number of the filters we used is z, for sentence S, we can get its representation $[u_q]_{q=1}^z$. Finally, after the input representation layer, the representations of the review text $\{\boldsymbol{S}_i^r\}_{i=1}^n$ and the abstract text $\{\boldsymbol{S}_j^a\}_{j=1}^m$ are denoted as $[\boldsymbol{I}_i]_{i=1}^n$, $[\boldsymbol{M}_j]_{i=1}^m$ respectively. where $\boldsymbol{I}_i, \boldsymbol{M}_j \in \mathbb{R}^z$.

Sentence Classification Layer: In this layer, we want to obtain new high-level representations of sentences in the review text by leveraging relevant abstract information. Given $\boldsymbol{I}_i$, which is the representation of $\boldsymbol{S}_i^r$, we can obtain a matched attention vector $\boldsymbol{E}^{(i)} = [e_t^{(i)}]_{t=1}^m$ which indicates the weight of memories. Then we calculate the response content $\boldsymbol{R}^{(i)} \in \mathbb{R}^z$ using this matched attention vector. It will be described in detail in the next section.

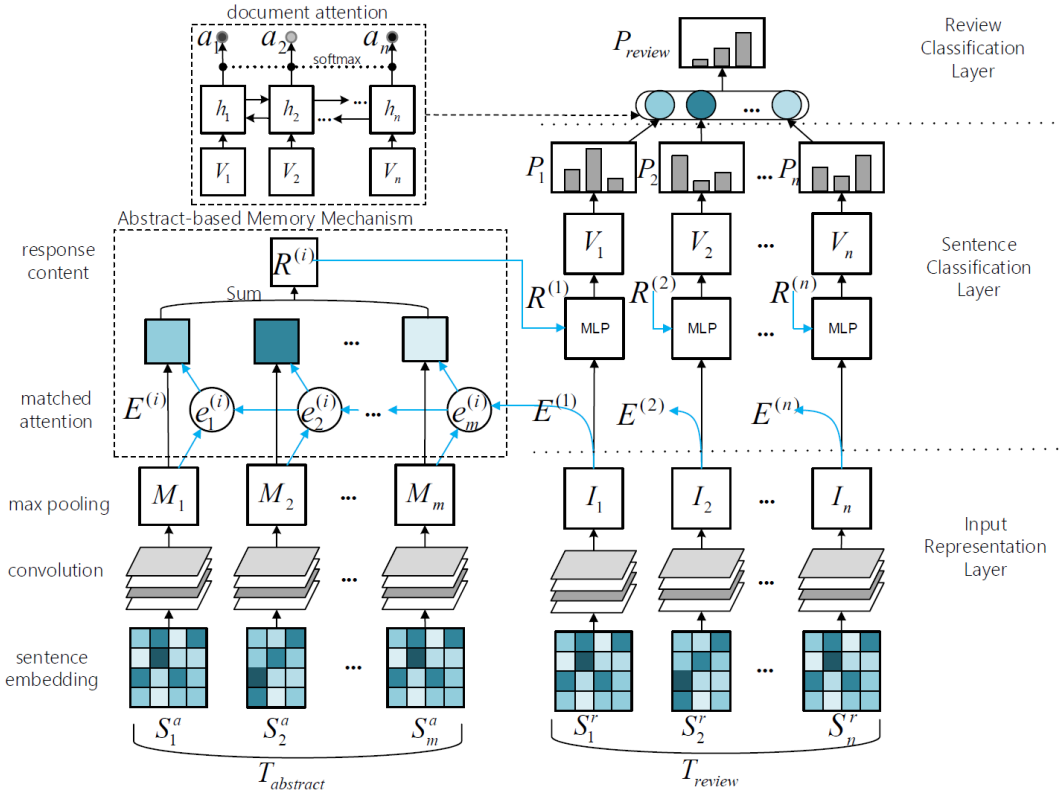

Figure 1: The architecture of our proposed multiple instance learning network with a novel abstract-based memory mechanism (MILAM). The inputs T_{review} and $T_{abstract}$ are a pair of review text and its corresponding abstract text.

After that, we use a multi-layer perceptron (MLP) to obtain the final representation vector of each sentence in the review text.

$$\boldsymbol{V}_i = f_{mlp}(\boldsymbol{I}_i || \boldsymbol{R}^{(i)}; \theta_{mlp}), \tag{5}$$

where || denotes the concatenation, f_{mlp} denotes one-hidden-layer-MLP, and θ_{mlp} denotes the parameters in it. The main motivation is we hope to leverage the abstract information of the paper to make the representation of the review to focus on the reviewer's own opinion and filter out the irrelevant noise.

Finally, we use the softmax classifier to get sentence-level distribution over sentiment labels.

$$\boldsymbol{P}_i = softmax(\boldsymbol{W}_p \cdot \boldsymbol{V}_i + \boldsymbol{b}_p), \tag{6}$$

where $\boldsymbol{W}_p$ and $\boldsymbol{b}_p$ are parameters, shared across all sentences. And $\boldsymbol{P}_i = [p_i^1, ..., p_i^C]$, where C is the total number of sentiment labels.

Review Classification Layer: In this layer, we predict the review's overall sentiment label based on the sentiment prediction results of sentences in it. As mentioned above, the sentiment label distribution of each sentence is $\boldsymbol{P}_i = [p_i^1, ..., p_i^C]$. Here we use a document attention mechanism which rewards sentences that are more likely to be good sentiment predictors.

We use separate LSTM modules to produce forward and backward hidden vectors, which are then concatenated:

$$\overrightarrow{h_i} = \overrightarrow{LSTM}(\boldsymbol{V_i}) \tag{7}$$

$$\overleftarrow{h_i} = \overleftarrow{LSTM}(\boldsymbol{V_i}) \tag{8}$$

$$h_i = \overrightarrow{h_i} || \overleftarrow{h_i} \tag{9}$$

The importance of each sentence is measured as follows:

$$h_i^{'} = tanh(\boldsymbol{W}_a \cdot h_i + \boldsymbol{b}_a) \tag{10}$$

$$a_i = \frac{exp(h_i^{'})}{\sum_j exp(h_j^{'})} \tag{11}$$

where Eq (10) defines a one-layer MLP that produces an attention weight a_i for the i-th sentence, and $\boldsymbol{W}_a$ and $\boldsymbol{b}_a$ are parameters in it.

Finally, we obtain a document-level distribution over sentiment labels as the weighted sum of sentence-level distributions:

$$P_{review}^{(c)} = \sum_i a_i P_i^{(c)}, c \in [1, C] \tag{12}$$

3.2 Abstract-based Memory Mechanism

Here we introduce our abstract-based memory mechanism in detail. In this study, with the intuition that the review text may have irrelevant noise (like repeating the contents of the paper) to confuse the classifier, we hope to leverage the abstract information to

control the representation of the review text , thus making it more focused on the reviewer's own opinions and improve the accuracy of sentiment classification.

However, the abstract text is usually long (contains multiple sentences) and it is difficult to be leveraged effectively by methods like hidden states or attention mechanism. Inspired by the memory network [26, 30, 35, 43], we treat the abstract information as memories, helping to control the representation of each sentence in the review text. And the abstract text representations $[\boldsymbol{M}_j]_{j=1}^{m}$ after the CNN layer can be considered as memory slots. For each sentence $\boldsymbol{I}_i$ in $[\boldsymbol{I}_i]_{i=1}^{n}$, the process of getting a new representation $\boldsymbol{V}_i$ can be divided into three steps:

Firstly, we get the matched attention vector $\boldsymbol{E}^{(i)}$ of memories. We use a layer of LSTM to obtain the $\boldsymbol{E}^{(i)}$ and use $\boldsymbol{I}_i$ to initialize the input.

$$e_t^{'} = LSTM(\hat{h}_{t-1}, \boldsymbol{M}_t), (\hat{h}_0 = \boldsymbol{I}_i, t = 1, ..., m) \tag{13}$$

$$e_t^{(i)} = \frac{exp(e_t^{'})}{\sum_j exp(e_j^{'})} \tag{14}$$

$$\boldsymbol{E}^{(i)} = [e_t^{(i)}]_{t=1}^{m} \tag{15}$$

where $\hat{h}_t$ is the hidden state of timestep t. $\boldsymbol{E}^{(i)}$ indicates the weights of memory slots (i.e., sentences of the abstract text).

Secondly, we calculate the response content $R^{(i)}$. We calculate the weighted sum of $[\boldsymbol{M}_t]_{t=1}^{m}$ as follows:

$$\boldsymbol{R}^{(i)} = \sum_{t=1}^{m} e_t^{(i)} \boldsymbol{M}_t \tag{16}$$

where $\boldsymbol{R}^{(i)}$ can be considered as a representation of the abstract text associated with sentence $\boldsymbol{I}_i$.

Finally, we use $\boldsymbol{R}^{(i)}$ and $\boldsymbol{I}_i$ to compute the new sentence representation vector $\boldsymbol{V}_i$ as mentioned earlier (Eq(5)).

3.3 Objective Function

Note that in the training dataset, we only have the gold-standard sentiment labels of reviews, but do not have the gold-standard sentiment labels of sentences. It means that our model only needs the review's sentiment label while each sentence's sentiment label is unobserved. Therefore, we use the categorical cross-entropy loss to minimize the sentiment prediction error between the output results and the gold-standard labels of reviews:

$$L(\theta) = \sum_{T_{review}} \sum_{c=1}^{C} -P_{review}^{(c)} \log(\bar{P}_{review}^{(c)}) \tag{17}$$

where T_{review} is the review text in the training data, $P_{review}^{(c)}$ and $\bar{P}_{review}^{(c)}$ are the true and predicted probabilities of belonging to the c-th class, respectively. We use Adam [21] with minibatch to learn the model parameter θ.

4 EXPERIMENTS

4.1 Evaluation Datasets

The peer reviews in most journals and conferences are not publicly available due to privacy issues. Fortunately, several conferences and workshops (e.g. ICLR, NIPS workshops, ICML workshops) have undergone open peer review and the peer reviews can be obtained on the OpenReview website [3]. However, only ICLR 2017 and ICLR 2018 provided both peer reviews and the corresponding overall recommendation scores for each submission, while other conferences and workshop only provide peer reviews without recommendation scores. So, in this study, we collected all peer review texts with recommendation scores from ICLR 2017 and ICLR 2018 as our evaluation datasets. A summary of statistics for each dataset is provided in Table 2. For each review text in them, an overall recommendation score was assigned accordingly. The recommendation scores range from 1 to 10 points, 10 means the best, and the score distributions of two datasets are shown in Figure 2.

Data Set	#Papers	#Reviews	#Sentences	#Words
ICLR-2017	490	1517	24497	9868
ICLR-2018	954	2875	58329	13503

Table 2: Statistics for ICLR-2017 and ICLR-2018 datasets.

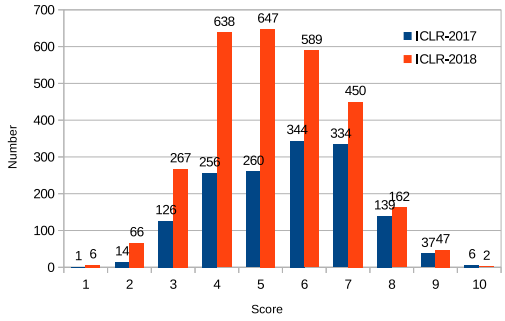

Figure 2: The score distributions of ICLR-2017 and ICLR-2018 peer reviews.

4.2 Experiment Setup

In this study, we do two classification tasks, one is **2-class** which predicts two classes: accept ($1 \leq score \leq 5$) and reject ($6 \leq score \leq 10$), the other is **3-class** which predicts three classes: accept ($1 \leq score \leq 4$), borderline ($5 \leq score \leq 6$) and reject ($7 \leq score \leq 10$). We use k-fold cross validation method ($k = 10$) to evaluate the model. Each dataset is randomly split into 10 parts, and at each fold we use one part for test and the other 9 parts for training. That is, at each fold, for ICLR-2017 dataset, we use 1366 reviews as training set, 151 reviews as test set; and for ICLR-2018 dataset, we use 2588 reviews as training set, 287 reviews as test set. The classification accuracy is calculated and averaged across 10 folds.

4.3 Implementation Details

We use pre-trained Google's 300-dimensional word vectors as initial word embeddings. The word vectors were fixed during the training process. The dropout rate [34] is set to 0.5 to prevent overfitting.

[3] https://openreview.net/

The number of CNN filters and LSTM hidden state units are set to 32 (z=32) and the sizes of MLP hidden units are all set to 32. For the hyperparameters of Adam optimizer, we set the learning rate $\alpha = 0.001$, two momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ respectively, and $\epsilon = 10^{-8}$. We implement our model based on Tensorflow and use a TITAN X graphic card for learning.

4.4 Baselines and Comparison Results

We compare our proposed model with several baseline methods, including some of state-of-art deep learning methods.

SVM(Uni): SVM has been widely used for sentiment classification in many text domains [27], so we simply use the unigrams in the review text as features and then use SVM to train a sentiment classifier. The feature weight is a binary value indicating the occurrence of the unigram. We also tried using frequency or TF-IDF as feature weights, but they are not as good as binary feature weights. Specifically, we use SVM implemented in python-sklearn[29].

SVM(Uni&Bi): We use the unigrams and bigrams in the review text as features, and then use SVM to train a sentiment classifier.

SVM(Uni&Bi&Senti): In this SVM method, we not only use unigrams and bigrams as features, but also extract several sentiment features, including the total number of sentiment words, the number of positive sentiment words, the number of negative sentiment words. We use SentiWordNet [13] as the sentiment dictionary to match sentiment words.

LSTM: LSTM (Long Short Term Memory) has achieved good performance for sentiment classification [16]. We use a one-layer LSTM network to obtain a high-level representation of the review text, and then use a softmax activation layer for sentiment classification.

CNN: CNN (Convolutional Neural Network) has also achieved excellent performance for sentiment classification [20]. We use CNN to get the high-level representation of the review text for classification.

CNN+Bi-LSTM: In this method, a hierarchical neural network is used. First of all, CNN is used to learn a presentation for each sentence in the review text, and then a bidirectional LSTM is used to obtain the high-level representation of the entire document based on the sentences' representations. We also tried other ways for combining LSTM and CNN (e.g. using LSTM for learning sentence-level representations and then using CNN for learning document-level representation), but the way used in this method performs best.

CNN+Bi-LSTM+Att: On the basis of the above CNN+LSTM model, we add a hierarchical attention mechanism [51]. The attention mechanism here includes the sentence-level attention and the document-level attention, and the attention scores are calculated based only on the review text in the same way as [51].

MIL: Here we use a state-of-art multiple instance learning framework [2] for comparison. It is noteworthy that the difference between our model and it is that we added a novel abstract-based memory mechanism.

We perform the same 10-fold cross validation to compare the performance of these methods on the two datasets. The average accuracy and the standard derivation for each method are shown in Table 3 and Table 4.

Methods	ICLR-2017	ICLR-2018
SVM(Uni)	70.14% (+/- 4.37%)	71.02% (+/- 3.81%)
SVM(Uni&Bi)	71.56% (+/- 2.52%)	73.23%(+/- 5.29%)
SVM(Uni&Bi&Senti)	74.23% (+/- 3.21%)	75.24% (+/- 2.36%)
CNN	68.93% (+/- 5.24%)	73.31% (+/- 2.70%)
LSTM	65.24% (+/- 4.21%)	69.25% (+/- 3.82%)
CNN+Bi-LSTM	74.35% (+/- 3.51%)	76.21% (+/- 4.08%)
CNN+Bi-LSTM+Att	75.64% (+/- 3.42%)	77.85% (+/- 3.72%)
MIL	75.02% (+/- 2.85%)	76.57% (+/- 2.31%)
MILAM	**78.24% (+/- 4.92%)**	**80.32% (+/- 5.43%)**

Table 3: Comparison of review sentiment classification accuracy on the 2-class task (accept, reject).

Methods	ICLR-2017	ICLR-2018
SVM(Uni)	45.75% (+/- 2.02%)	50.25% (+/- 3.83%)
SVM(Uni&Bi)	48.79% (+/- 3.50%)	54.11% (+/- 4.16%)
SVM(Uni&Bi&Senti)	50.44% (+/- 5.36%)	55.37% (+/- 4.35%)
CNN	49.65% (+/- 5.86%)	55.59% (+/- 5.28%)
LSTM	44.24% (+/- 3.54%)	51.25% (+/- 4.24%)
CNN+Bi-LSTM	53.50% (+/- 4.31%)	57.04% (+/ 3.63%)
CNN+Bi-LSTM+Att	55.91% (+/- 5.29%)	60.27% (+/ 3.24%)
MIL	56.15% (+/- 4.11%)	59.35% (+/- 3.85%)
MILAM	**61.29% (+/- 4.82%)**	**62.64% (+/- 5.15%)**

Table 4: Comparison of review sentiment classification accuracy on the 3-class task (accept, borderline, reject).

From the comparison results in the tables we can see that our proposed model outperforms all baseline methods. The average accuracy achieved by our model is promisingly high, reaching 78.24% and 80.32% over the two datasets on the 2-class task, and exceeding 60% over the two datasets on the 3-class task. For SVM methods, both the use of the bigram features and the use of sentiment features are beneficial to the classification performance, because they can capture more sentiment information. Interestingly, the basic LSTM and CNN methods do not perform well, while the use of attention mechanism and hierarchical architecture can much improve the performance, which demonstrate that the attention mechanism and hierarchical architecture can be used to learn a better document representation for sentiment classification.

In particular, our proposed model(MILAM) performs much better than MIL, and it demonstrates the great helpfulness of our abstract-based memory mechanism. Moreover, compared with CNN+Bi-LSTM+Att, MILAM can predict the sentiment label of each sentence at the same time, which not only makes the neural network interpretable, but also extracts opinions of the reviewer.

4.5 Sentence-Level Classification Results

Here we want to verify that our model can accurately predict the sentiment polarities of sentences in the reviews. It is worth noting that we do not have the sentiment label of each sentence, so we randomly selected 20 reviews, a total of 213 sentences, and manually labeled the sentiment polarity of each sentence, and used them as test set. We compare our model and the MIL model on this test

#	Polarity			Review Sentences
1	Human	MIL	MILAM	
2	+	+0.18	+0.10	This paper introduces a novel hierarchial memory architexture for neural networks, based on a binary tree with leaves corresponding to memory cells.
3	+	+0.13	+0.09	This allows for O(log n) memory access, and experiments additionally demonstrate ability to solve more challenging tasks such as sorting from pure input-outputexamples and dealing with longers sequences.
4	+	+0.21	+0.15	The idea of the paper is novel and well-presented, and the memory structure seems reasonable to have advantages in pratice.
5	-	-0.15	-0.31	However, the main weakness of the paper is the experiments.
6	-	+0.12	-0.21	There is no experimental comparison with other external memory-based approaches (e.g. those discussed in Related Work), or experimental analysis of computational efficiency given overhead costs (beyond just computational complexity) despite that being one of the main advantages.
7	-	+0.07	-0.15	Furthermore, the experimental setups are relatively weak, all on artificial tasks with moderate increases in sequence length.
8	-	+0.14	-0.11	Improving on these would greatly strengthen the paper, as the core idea is interesting.

Table 5: Example opinionated sentences with predicted polarity scores extracted from a review text. The real values indicates polarity scores, + means positive opinion, and − means negative opinion.

Method	Accuracy
MIL	72.97%
MILAM	81.12%*

Table 6: Accuracy on sentence-level sentiment classification. $*$ means the performance improvement over MIL is statistically significant with p-value < 0.05 for sign-test.

set, as shown in Table 6. Note that the MIL model can predict the sentence-level sentiment labels as our model, while other baseline methods cannot do this.

As can be seen from Table 6, MILAM much outperforms MIL on sentence-level sentiment classification, which demonstrates that the introduction of abstract-based memory mechanism can greatly improve the performance. Further, we want to use the sentence-level predictions to extract opinion summaries. We introduce a method that takes our model's confidence in the prediction into account, by reducing each sentence's class probability distribution P_i to a single real-valued polarity score. To achieve this, we first define a real-valued class weight vector $\boldsymbol{W} = [W^{(i)}]_{i=1}^{C}, W^{(i)} \in [-1, 1]$ that assigns uniformly-spaced weights to the ordered label-set, such that $W^{(i+1)} - W^{(i)} = \frac{2}{C-1}$. For example, in the 2-class scenario, the class weight vector would be $W = [-1, 1]$. We compute the polarity score of a segment as the dot-product of the probability distribution P_i with vector W:

$$polarity(S_i^r) = \sum_{j=1}^{C} P_i^{(j)} W^{(j)} \in [-1, 1] \tag{18}$$

Table 5 show example opinionated sentences with polarity scores extracted from a review text. From the table, we can see that MILAM performs better than MIL in terms of the accuracy of polarity labels (See rows 6-8), and more reasonably in assigning polarity scores (See rows 2-5), especially when there is much paper content in the sentence.

In general, MILAM much outperforms MIL in sentence-level sentiment classification, and thus we can extract opinions of the reviewer based on the predicted polarity scores of sentences, which are convenient for authors to further improve their paper.

Methods	ICLR-2017		ICLR-2018	
	original	remove abstract	original	remove abstract
SVM (Uni&Bi)	71.56% (+/- 2.52%)	71.57% (+/- 4.12%)	73.23% (+/- 5.29%)	73.42% (+/- 3.24%)
SVM (Uni&Bi&Senti)	74.23% (+/- 3.21%)	**75.16% (+/- 3.12%)**	75.24% (+/- 2.36%)	**75.76% (+/- 4.62%)**
CNN+ Bi-LSTM+Att	75.64% (+/- 3.42%)	75.74% (+/- 2.42%)	77.85% (+/- 3.72%)	77.65% (+/- 3.35%)
MIL	75.02% (+/- 2.85%)	75.12% (+/- 2.83%)	76.57% (+/- 2.31%)	76.55% (+/- 3.15%)

Table 7: The comparison of using and not using the paper abstract via a simple method.

4.6 Influence of Abstract Text

Here we want to investigate the influence of the abstract text in sentiment classification. We first observe the matched attention vector $E^{(i)}$ of each sentence, and we consider the abstract's sentence with the largest weight in it as the most relevant sentence for a review sentence. Table 8 shows the results.

In the table, these most relevant sentences are likely to be descriptions of the content of the paper and they are not useful for sentiment classification. So we design a simple method of using abstract texts as a contrast experiment. That is, we directly remove the sentences that are similar to the paper abstract's sentences from the review text and use the remaining text for classification. The sentence similarity is measured by the Overlap Similarity and the threshold of sentence removal is set to 0.7.

The comparison results on the 2-class task are shown in Table 7. In the table, methods performed on the original review texts are denoted as "original", and methods performed on the remaining review texts after sentence removal are denoted as "remove abstract". The results show that this simple method of using abstract texts can

Review Sentences	Abstract Sentences
This paper introduces a novel hierarchical memory architecture for neural networks, based on a binary tree with leaves corresponding to memory cells.	In this paper, we propose and investigate a novel memory architecture for neural networks called Hierarchical Attentive Memory (HAM).
This allows for O(log n) memory access, and experiments additionally demonstrate ability to solve more challenging tasks such as sorting from pure input-output examples and dealing with longer sequences.	This allows HAM to perform memory access in O(log n) complexity, which is a significant improvement over the standard attention mechanism that requires O(n) operations, where n is the size of the memory.
The idea of the paper is novel and well-presented, and the memory structure seems reasonable to have advantages in practice.	We also show that HAM can be trained to act like classic data structures: a stack, a FIFO queue and a priority queue.
...	...

Table 8: Example sentences in a review text and its most relevant sentence in the paper abstract text. The sentence with the largest weight in the matched attention vector $E^{(i)}$ is considered most relevant. The red texts indicate similarities in the review text and the abstract text.

Methods	ICLR-2017			ICLR-2018		
	w/o borderline reviews	w/ borderline reviews	only borderline reviews	w/o borderline reviews	w/ borderline reviews	only borderline reviews
SVM (Uni&Bi)	78.25% (+/- 2.77%)	71.56% (+/- 2.52%)	55.74% (+/- 4.92%)	79.25% (+/- 4.17%)	73.23% (+/- 5.29%)	56.10% (+/- 6.49%)
SVM (Uni&Bi&Senti)	80.21% (+/- 4.90%)	74.23% (+/- 3.21%)	56.67% (+/- 3.77%)	80.89% (+/- 2.65%)	75.24% (+/- 2.36%)	56.65% (+/- 3.28%)
CNN+ Bi-LSTM+Att	85.62% (+/- 2.62%)	74.35% (+/- 3.51%)	62.30% (+/- 3.78%)	87.21% (+/- 2.36%)	77.85% (+/- 3.72%)	63.71% (+/- 5.74%)
MIL	85.43% (+/- 3.31%)	75.02% (+/- 2.85%)	61.67% (+/- 5.24%)	86.29% (+/- 3.66%)	76.57% (+/- 2.31%)	63.07% (+/- 6.56%)
MILAM	**89.23% (+/- 4.14%)**	**78.24% (+/- 4.92%)**	**63.17% (+/- 5.24%)**	**90.64% (+/- 5.54%)**	**80.32% (+/- 5.43%)**	**65.26% (+/- 4.75%)**

Table 9: Experimental results on different datasets with, without and only borderline reviews.

be a minor improvement on SVM (Uni&Bi&Senti), but the abstract text can not be used effectively as our method does. We guess that simply removing highly similar sentences will also remove some of the useful information.

4.7 Influence of Borderline Reviews

When we do the 3-class task, we define the borderline reviews as having a score of 5 or 6. In this section, we further investigate the influence of borderline reviews. We perform 2-class sentiment classification on three different kinds of datasets: the full datasets containing borderline reviews (we label a review of score 1~5 as reject and 6~10 as accept), the datasets after removing borderline reviews (1~4 as reject and 7~10 as accept) , and the datasets that contain only borderline reviews (5 as reject and 6 as accept). The average accuracy of different methods (including our model and four strong baseline methods) on different datasets is shown in Table 9.

From Table 9, we can see that the performance on datasets after removing borderline reviews is largely improved, especially in deep learning methods, and the performance on datasets that contain only borderline reviews is very low. All of above indicate that the borderline reviews are hard to be differentiated because reviewers usually write similar texts and express similar opinions in borderline reviews. Except for borderline reviews, the other review texts are generally in good consistency with their overall recommendation. It is noteworthy that no matter which dataset is used, our proposed model (MILAM) performs best.

Test Dataset	ICLR-2017		ICLR-2018	
Model	Model@ICLR-2018		Model@ICLR-2017	
Task	2-class	3-class	2-class	3-class
SVM(Uni&Bi)	70.24%	47.01%	70.13%	50.26%
SVM(Uni&Bi&Senti)	74.34%	49.76%	73.37%	52.73%
LSTM+CNN+Att	75.01%	55.65%	76.32%	55.76%
MIL	75.35%	55.25%	75.26%	55.27%
MILAM	**79.35%**	**61.02%**	**78.92%**	**61.27%**

Table 10: Results of cross-year experiments. *Model@ICLR − ∗* **means the model is trained on** *ICLR − ∗* **dataset.**

4.8 Cross-Year Experiments

Further, we do a cross-year experiment. We use the model trained on the ICLR-2017 dataset to test on ICLR-2018 dataset, and vice versa. The results is shown in Table 10. We can see that our model still

Methods	Accuracy	
	Model@ICLR-2017	Model@ICLR-2018
SVM(Uni&Bi)	63.54%	64.24%
SVM(Uni&Bi&Senti)	66.72%	66.98%
LSTM+CNN+Att	72.23%	73.51%
MIL	71.37%	73.21%
MILAM	**76.83%***	**78.10%***

Table 11: Results of cross-domain experiments.$*$ means the performance improvement over the first three methods is statistically significant with p-value < 0.05 for sign-test. *Model@ICLR* $-$ $*$ means the model is trained on *ICLR* $-$ $*$ dataset.

Methods	Voting		Simple Average		Confidence-based Average	
	2-class	3-class	2-class	3-class	2-class	3-class
SVM (Uni&Bi)	62.24%	67.84%	60.62%	64.36%	55.37%	56.36%
SVM (Uni&Bi&Senti)	65.26%	69.47%	63.92%	66.72%	58.47%	60.37%
CNN+Bi-LSTM+Att	76.35%	78.25%	74.18%	75.53%	63.46%	65.86%
MIL	75.36%	78.53%	73.29%	74.90%	63.29%	64.89%
MILAM	79.86%	**81.69%**	78.25%	79.65%	66.25%	69.02%

Table 12: Results of final decision prediction for scholarly papers.

maintains a good performance and it demonstrates the robustness of our proposed model.

4.9 Cross-Domain Experiments

In the above experiments, we used the ICLR datasets which focuses on machine learning techniques and applications. We further collected 87 peer reviews (after removing borderline reviews) for submissions in the Natural Language Processing conferences (CoNLL, ACL, EMNLP, etc.), including 57 positive reviews (accept) and 30 negative reviews (reject). The 87 peer reviews are used as another test set in a different domain. We performed cross-domain classification experiments by applying the models learned on two ICLR datasets to the test set in the NLP field. The results are shown in Table 11 and our model still performs much better than the four strong baseline methods. The performance of our model do not decrease much, even though the cross-domain difference between the training and test reviews. The cross-domain adaptation problem is a typical and difficult problem for text classification, and advanced transfer learning techniques may be used to address this problem in our future work.

4.10 Final Decision Prediction for Scholarly Papers

Further, we investigate whether the above prediction results can be used to predict the final decision of accepting/rejecting a paper based on several (usually 3) reviews for the paper. However, until we write this paper, the final results of the papers in ICLR-2018 have not been published yet. So we only test on the ICLR-2017 dataset. We designed three methods to predict the final decision of a paper based on several review scores.

Voting: A simple voting method is used on the basis of the predicted labels of review texts to determine the final decision.

$$Decision = \begin{cases} Accept & \text{if \#accept > \#reject} \\ Reject & Otherwise \end{cases} \quad (19)$$

where #accept means the number of reviews that are predicted as accept, and so does #reject.

Simple Average: In this method, we obtain the predicted probability score for the accept class of each review , and then simply average the scores of all reviews. If the average score is larger than or equal to 0.6, then the paper is predicted as final accept, and otherwise final reject.

Confidence-based Average: Different from the simple average method, we use the reviewer' confidence score to compute a weight for the predicted probability score, and use the weighted average for determination.

$$overall\_score = \frac{1}{|S|}\sum_{i=1}^{|S|} S_i * \frac{1}{(6 - ReviewerConfidence_i)} \quad (20)$$

where S_i is the predicted probability score for the accept class of each review and S is the set of scores for a paper. $ReviewerConfidence_i$ ranges from 1 to 5. If the weighted average is larger than or equal to 0.4, then the paper is predicted as final accept, and otherwise final reject.

Table 12 shows our experimental results. From the table, we can see that Confidence-based Average method is not good, and the Voting method based on 3-class classification is the best choice to predict the final decision. Moreover, our proposed model always much outperform the baseline methods for final decision prediction for papers in different settings.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed a neural network model with a novel abstract-based attention mechanism to address the challenging task of sentiment analysis in the domain of peer reviews for scholarly papers. We constructed two evaluation datasets from the ICLR open reviews and discussed evaluation results extensively. We verified the efficacy of our proposed model, which achieved high accuracy in different settings. We found that the consistency between the peer review texts and the recommended decisions for ICLR was generally good, except for borderline reviews. Moreover, our model can extract reviewers' opinions at the same time, which brings convenience for authors to further improve their papers.

In future work, we will collect more peer reviews in multiple research areas for training and test, and we will try more sophisticated deep learning techniques to address this task. We will also investigate transfer learning techniques to address the domain adaptation problem.

Several other sentiment analysis tasks in this domain are also interesting and worth of investigation, e.g., prediction of the fine-granularity scores of reviews, automatic writing of meta-reviews, prediction of the best papers, and so on.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (61772036, 61331011) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We thank the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

REFERENCES

[1] Stuart Andrews and Thomas Hofmann. 2004. Multiple Instance Learning via Disjunctive Programming Boosting. (2004), 65–72.
[2] Stefanos Angelidis and Mirella Lapata. 2017. Multiple Instance Learning Networks for Fine-Grained Sentiment Analysis. *CoRR* abs/1711.09645 (2017). arXiv:1711.09645 http://arxiv.org/abs/1711.09645
[3] Dale J Benos, E Bashari, J M Chaves, A Gaggar, N Kapoor, M Lafrance, R Mans, D Mayhew, S Mcgowan, A Polter, et al. 2007. The ups and downs of peer review. *Advances in Physiology Education* 31, 2 (2007), 145–152.
[4] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification.. In *ACL*. 187–205.
[5] Antoine Bordes, Ylan Boureau, and Jason Weston. 2016. Learning End-to-End Goal-Oriented Dialog. *arXiv: Computation and Language* (2016).
[6] Kwangsu Cho. 2008. Machine Classification of Peer Comments in Physics. In *Educational Data Mining 2008, the International Conference on Educational Data Mining, Montreal, QuÃlbec, Canada, June 20-21, 2008. Proceedings*. 192–196.
[7] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* 12 (2011), 2493–2537.
[8] Timothee Cour, Benjamin Sapp, and Ben Taskar. 2011. Learning from Partial Labels. *Journal of Machine Learning Research* 12 (2011), 1501–1536.
[9] Kushal Dave, David M. Pennock, and David M. Pennock. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *International Conference on World Wide Web*. 519–528.
[10] Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging Large Amounts of Weakly Supervised Data for Multi-Language Sentiment Classification *(WWW '17)*. Republic and Canton of Geneva, Switzerland, 1045–1052. https://doi.org/10.1145/3038912.3052611
[11] Thomas G Dietterich, Richard H Lathrop, and Tomas Lozanoperez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 1 (1997), 31–71.
[12] Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander H Miller, Arthur Szlam, and Jason Weston. 2016. Evaluating Prerequisite Qualities for Learning End-to-End Dialog Systems. *international conference on learning representations* (2016).
[13] Andrea Esuli and Fabrizio Sebastiani. 2006. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. (2006).
[14] Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *Cs224n Project Report* (2009).
[15] Namrata Godbole, Manjunath Srinivasaiah, and Steven Skiena. 2007. Large-Scale Sentiment Analysis for News and Blogs. In *International Conference on Weblogs and Social Media*.
[16] Alex Graves. 2012. *Supervised Sequence Labelling with Recurrent Neural Networks*. Studies in Computational Intelligence, Vol. 385. Springer. https://doi.org/10.1007/978-3-642-24797-2
[17] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S Zettlemoyer, and Daniel S Weld. 2011. Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations. (2011), 541–550.
[18] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter sentiment classification. In *Meeting of the Association for Computational Linguistics: Human Language Technologies*. 151–160.
[19] James David Keeler and David E Rumelhart. 1992. A Self-Organizing Integrated Segmentation and Recognition Neural Net. (1992), 496–503.
[20] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. *Eprint Arxiv* (2014).
[21] Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. *international conference on learning representations* (2015).
[22] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From Group to Individual Labels Using Deep Features. (2015), 597–606.
[23] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: dynamic memory networks for natural language processing. *international conference on machine learning* (2016), 1378–1387.
[24] Nana Li, Shuangfei Zhai, Zhongfei Zhang, and Boying Liu. 2016. Structural Correspondence Learning for Cross-lingual Sentiment Classification with One-to-many Mappings. *national conference on artificial intelligence* (2016), 3490–3496.
[25] Oded Maron and Aparna Lakshmi Ratan. 1998. Multiple-Instance Learning for Natural Scene Classification. (1998), 341–349.
[26] Alexander H Miller, Adam Fisch, Jesse Dodge, Amirhossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-Value Memory Networks for Directly Reading Documents. *empirical methods in natural language processing* (2016), 1400–1409.
[27] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *empirical methods in natural language processing* (2002), 79–86.
[28] Nikolaos Pappas and Andrei Popescubelis. 2014. Explaining the Stars: Weighted Multiple-Instance Learning for Aspect-Based Sentiment Analysis. (2014), 455–466.
[29] 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 (2011), 2825–2830.
[30] Julien Perez and Fei Liu. 2017. Gated End-to-End Memory Networks. *conference of the european chapter of the association for computational linguistics* (2017), 1–10.
[31] Lakshmi Ramachandran and Edward F. Gehringer. 2011. Automated Assessment of Review Quality Using Latent Semantic Analysis. In *IEEE International Conference on Advanced Learning Technologies*. 136–138.
[32] Y Ren, Y Zhang, M Zhang, and D Ji. 2016. Context-sensitive twitter sentiment classification using neural network. (01 2016), 215–221.
[33] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. (2013), 1631–1642.
[34] 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 (2014), 1929–1958.
[35] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Robert D Fergus. 2015. End-to-end memory networks. *neural information processing systems* (2015), 2440–2448.
[36] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics* 37, 2 (2011), 267–307.
[37] Oscar Tackstrom and Ryan T Mcdonald. 2011. Discovering fine-grained sentiment with latent variable structured prediction models. (2011), 368–374.
[38] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. (2015), 1422–1432.
[39] Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning Semantic Representations of Users and Products for Document Level Sentiment Classification. In *Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing*. 1014–1023.
[40] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect Level Sentiment Classification with Deep Memory Network. *empirical methods in natural language processing* (2016), 214–224.
[41] Peter D. Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL*. 417–424.
[42] Jason Weston. 2016. Dialog-based Language Learning. *neural information processing systems* (2016), 829–837.
[43] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. *CoRR* abs/1410.3916 (2014). arXiv:1410.3916 http://arxiv.org/abs/1410.3916
[44] Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. 2015. Deep multiple instance learning for image classification and auto-annotation. (2015), 3460–3469.
[45] Wenting Xiong and Diane Litman. 2011. Automatically predicting peer-review helpfulness. In *Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers*. 502–507.
[46] Wenting Xiong, Diane J. Litman, and Christian D. Schunn. 2010. Assessing Reviewers ' Performance Based on Mining Problem Localization in Peer-Review Data. In *Educational Data Mining 2010, the International Conference on Educational Data Mining, Pittsburgh, Pa, Usa, June 11-13, 2010. Proceedings*. 211–220.
[47] Wei Xu, Alan Ritter, Chris Callisonburch, William B Dolan, and Yangfeng Ji. 2014. Extracting Lexically Divergent Paraphrases from Twitter. *Transactions of the Association for Computational Linguistics* 2, 0 (2014), 435–448.
[48] Ravi Yadav and Edward F. Gehringer. 2016. Automated Metareviewing: A Classifier Approach to Assess the Quality of Reviews. In *Workshop and Tutorial Proceedings of the 9th International Conference on Educational Data Mining, Raleigh, NC, USA, June 29, 2016.*
[49] Cha Zhang, John Platt, and Paul A Viola. 2006. Multiple Instance Boosting for Object Detection. (2006), 1417–1424.
[50] Qi Zhang, Sally A Goldman, Wei Yu, and Jason E Fritts. 2002. Content-Based Image Retrieval Using Multiple-Instance Learning. (2002), 682–689.
[51] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Attention-based LSTM Network for Cross-Lingual Sentiment Classification. In *EMNLP*. 247–256.