

PAPER • OPEN ACCESS

A GRU Model for Aspect Level Sentiment Analysis

To cite this article: Yongping Xing and Chuangbai Xiao 2019 *J. Phys.: Conf. Ser.* **1302** 032042

View the [article online](#) for updates and enhancements.



IOP | ebooks™

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

A GRU Model for Aspect Level Sentiment Analysis

Yongping Xing and Chuangbai Xiao

Department of Computer Science and Technology, Beijing University of Technology,
Beijing, China

hsingyp@163.com, cbxiao@bjut.edu.cn

Abstract. Sentiment analysis is a basic task of natural language processing, while aspect level sentiment analysis is an important topic in sentiment analysis. In the same sentence, different words have different influence on the sentiment polarity of aspect, so the key to solve the problem is how to build a relation model between the aspect and the words in the sentence. In this paper, by using two recurrent networks, we built a model for sentence and introduced attention mechanism to fuse aspect information, so as to achieve a better effect. An experiment on public dataset show that the proposed algorithm obtain a better result without carrying out complex feature engineering.

1. Introduction

With the rapid development of Internet, there are more and more text messages. So it becomes increasingly important how to obtain useful information from the massive text messages. At the same time, those massive text messages objectively promote the development of natural language processing. In natural language processing, sentiment analysis is a fundamental but very important task [1, 2]. As personal emotion may have significant impact on human's behavior and activity, therefore sentiment analysis has important application in business, politics and other fields [2]. Enterprises can take advantage of customers' comments on products to obtain immediate response so as to provide reference for decision-making. Consequently, in recent years, how to extract emotive information from massive text data has turned to a significant research subject in natural language processing.

So far, the text sentiment analytical research is mainly based on sentiment lexicons and machine learning. Method based on sentiment lexicons relies on sentiment lexicons, so sentiment lexicon can greatly affect sentiment analysis [3, 4]. [5] Processed and expressed texts on basis of sentiment lexicons, thus constructing a Naive Bayesian Theory based classifier. The other method is based on machine learning. In the machine learning method, the manually calibrated data are trained to obtain a sentiment. Analysis classifier [6, 7], and experiments are made to prove the excellent classification performance of support vector machine. Methods based on sentiment lexicons and machine learning need manually calibrated data to compete the construction of sentiment lexicons and feature engineering, which are tedious and complicated tasks. But the deep learning algorithm can solve this problem fairly well. In recent years, deep learning has achieved great success in natural language processing, such as machine translation [8], question-answering system [9], and named-entity recognition [13]. It is also applied in sentiment analysis field. Socher et al.[10] put forward a semi-supervised recursive auto encoder(RAE) based deep learning method to realize text sentiment classification; Jurgovsky et al.[11] used convolutional neural network(CNN) to achieve text sentiment classification. The text sentiment analysis can be classified into text level, sentence level and aspect level. This paper mainly studied aspect based sentiment analysis. For example, in the sentence "The voice quality of this phone is not good, but the battery life is long", this sentence gives a negatively evaluation for "quality", but a positive evaluation for "battery life". This is because the emotional polarity of different aspects in the same sentence could



be different. Wang et al. [12] proposed AE-LSTM, AT-LSTM and AEAT-LSTM algorithms on the aspect sentiment analysis, and fused aspect information into long-term and short-term memory network so as to improve classification accuracy. SVM-dep algorithm [7] divided features into aspect-related features and aspect-irrelated features. The aspect level sentiment analysis is completed by such division, in which accuracy is better than SVM classifier without aspect features. AdaRNN-w/E and AdaRNNcomb are based on Adaptive Recursive Neural Network (AdaRNN), which classify the sentence based on dependency parsing or unigram/bigram features [16]. AB-LSTM [21] is derived from attention-based bidirectional LSTM. It models the preceding and following contexts using bidirectional LSTM, and then attention is placed on bidirectional LSTM, which considers target information.

Attention mechanism is a kind of information processing mechanism, which selectively focuses on certain important information when processing information, ignoring the information that has weak significance correlation with focused target. It emphasizes on paying more attention to the nature of the information in information processing, and concentrates limited resources on the processing of important information, thereby achieving great success. Attention mechanism has acquired big achievements in image processing [17, 18], natural language processing [14, 19]. Combined with the theme of this paper, when processing the aspect level sentiment analysis, we could pay more attention to aspect-related information so as to increase the accuracy of sentiment classification.

Due to its network memory performance to treat contextual information, recurrent neural network (RNN) is widely used in natural language processing. Typical RNN includes LSTM, GRU. The paper will propose a GRU based aspect granular sentiment analysis algorithm, which fuses aspect information into the model through attention mechanism, making the algorithmic model to concern more of the influence of aspect on sentiment classification, so as to improve sentiment classification accuracy.

2. GRU Recurrent Network

GRU is a variant of recurrent network. Like LSTM, it can overcome the gradient vanishing problem of traditional RNN. This is because it introduces adaptive gate mechanism, which enables GRU network unit to not only process current data information but also effectively utilize previous data information. This function is completed by reset gate unit r_t and update gate unit z_t together. The forward propagation of GRU recurrent network is denoted as below:

$$r_t = \sigma(W_{rh}h_{t-1} + W_{rx}x_t + b_r) \quad (1)$$

$$z_t = \sigma(W_{zh}h_{t-1} + W_{zx}x_t + b_z) \quad (2)$$

$$\tilde{h}_t = \tanh(W_{hh}(r_t * h_{t-1}) + W_{hx}x_t + b_h) \quad (3)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (4)$$

$$o_t = \sigma(W_o \cdot h_t) \quad (5)$$

In which x_t is the input at the moment t .

3. Att-CGRU Algorithm(Attention Connect GRU Algorithm)

The Att-CGRU model will be introduced in detail in this part, see its structure in figure 1.

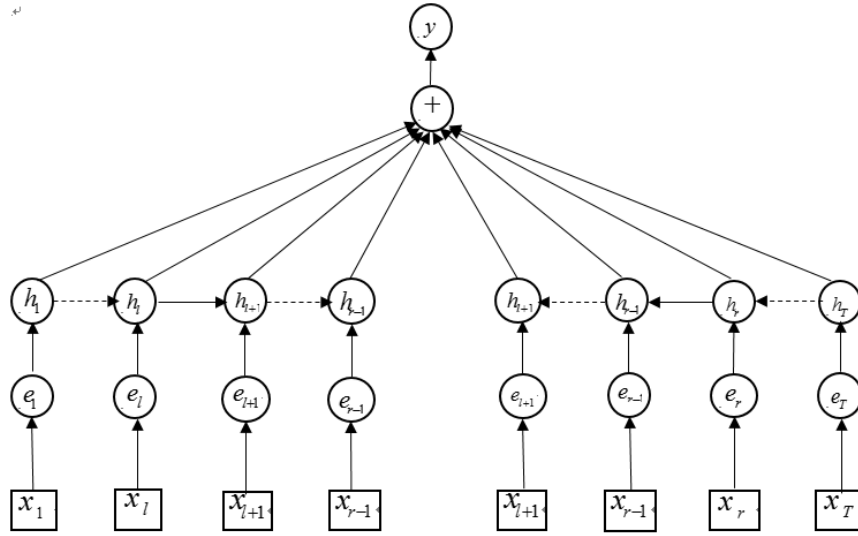


Figure 1. Architecture of Att-CGRU

The model includes five parts: input layer, embedded layer, GRU layer, attention layer and output layer. The input layer inputs a short text, i.e. a sentence, in the model; the embedded layer maps each word in the sentence into a vector; GRU layer utilized word embedding to get feature information; attention layer realizes the attention mechanism, which fuses the word level feature information into sentence level feature information through weight calculation, so as to produce a sentence feature vector; finally, the sentence feature vectors are classified.

3.1. Embedded Layer

If one input sentence includes T terms, then this sentence can be expressed as $s = \{x_1, x_2, \dots, x_T\}$, and each word x_i corresponds to one word vector emb_i . First, we obtain the word vector of each word from the word embedded matrix $W^{wrd} \in \mathbb{R}^{d^w \times |V|}$, in which, V is the length of word list, d^w is the dimension of word vector that can be assigned, then

$$emb_i = W^{wrd} v^i \quad (6)$$

In which, v^i is a vector with the length of $|V|$, which equals to 1 at location i and equals to zero at other positions. Similarly, we can get the word vector emb_{asp} of aspect. When the aspect in the sentence is multiple words, we add the value of word vector of each word at the same dimension and obtain the word vector of aspect. Then, we put emb_i and emb_{asp} together and obtain e_i

$$e_i = [emb_i : emb_{asp}] \quad (7)$$

Finally, $e = \{e_1, e_2, \dots, e_T\}$ is input in the next layer.

3.2. GRU Layer

We will take aspect as the cut-off point, and divide the sentence into a left part and a right part to build a model for aspect context in GRU layer. $\{x_{l+1}, x_{l+2}, \dots, x_{r-1}\}$ Denotes to aspect, $\{x_1, x_2, \dots, x_l\}$ represents the words before aspect in the sentence, and $\{x_r, x_{r-2}, \dots, x_T\}$ represents the words after aspect. After inputting the left sequence and the right sequence in the left and right networks, the hidden layer can obtain $\{h_1, h_2, \dots, h_{r-1}\}$ and $\{h_{l+1}, h_{l+2}, \dots, h_T\}$ respectively.

3.3. The Attention Layer in Att-CGRU Model

In the Att-CGRU model, we introduced the attention mechanism to obtain a better classification effect. This is because the different words and aspects in the front and back parts of the sentence have different relations. We will pay more attention to the information closely related to aspect. Attention mechanism is implemented by the following formula

$$M = \tanh\left(\begin{bmatrix} W_h H \\ W_v e_{asp} \otimes e \end{bmatrix}\right) \quad (8)$$

$$a_t = \text{soft max}(w^T M) \quad (9)$$

$$r = H a_t \quad (10)$$

Where a_t denotes weight in attention mechanism, \otimes is repeating the linearly transformed e_{asp} as many times as the width of H , H is the output of GRU hidden layer, r denotes vector representation of sequences, W_h , W_v , w are the parameter matrices. Then we get the vector representation o of the input sentence through the following formula

$$o = \tanh(W_p r + W_x h) \quad (11)$$

h is sum of h_{t-1} and h_{t+1}

Then we predict the label through the following formula

$$\hat{y} = \text{soft max}(W_o o + b_o) \quad (12)$$

Where softmax is the softmax function, W_o and b_o are the parameter matrices.

4. Experiment and Analysis

4.1. Model Training

The model is trained by minimising the loss function, where cross-entropy loss is regarded as the loss function. Given a training sample x , y is the ground truth distribution for the sentence and \hat{y} is the predicted sentiment distribution as follows:

$$\text{loss} = \sum_i \sum_j y_i^j \log(\hat{y}_i^j) + \lambda \|\theta\|^2 \quad (13)$$

Where j is the index of the class, which is positive, neutral or negative, i is the index of the sentence; λ is the L_2 -regularisation term and θ is the parameter set. A dropout layer is imposed on the penultimate layer with a constraint on the L2-norms of the weight vectors. The dropout layer can randomly set hidden units to 0 during forward-backward propagation with a probability of p .

4.2. Experiment Settings

In this work, the experiment is conducted on an original dataset [16] collected from Twitter, where every instance is manually labelled sentiment polarity. The dataset includes a training set that contains 6,248 sentences and a test set that contains 692 sentences. Positive, neutral and negative sentences in the training and test datasets have the same percentages of 25%, 50% and 25%, respectively. Non-alphabet characters, punctuation marks, numbers and stop words are removed from the sentences during preprocessing. Accuracy over positive, neutral and negative categories are regarded as the evaluation metrics for our model.

The word embedding vectors of the input sentence are initialised using Glove [20], where the word embedding vectors are pretrained on an unlabelled Twitter corpus. The dimensions of the word embedding vectors and the aspect term word embedding vectors are set to 200. When a word that is not present in the set of pretrained words is encountered, it is initialised randomly. The other model

parameters are initialised by sampling from a uniform distribution. L2-regularisation is implemented on the softmax layer with a weight of 0.01. The models are trained with a batch size of 32 examples, and AdaGrad is adopted as the optimisation method. This method remarkably improves the robustness of the stochastic gradient descent on large-scale learning tasks and initialised the learning rate by 0.001.

4.3. Experiment and Analysis

In the experiment, we adopted accuracy value as the model performance index. We compared it with relevant models, such as SVM [6], SVM-dep [7], AdaRNN-w/E, AdaRNNcomb [16] algorithm and AB-LSTM [21]. SVM takes n-gram information, punctuation mark, emoticon, positive and negative word quantity and so on as its feature; SVM-dep also takes target information as its classification feature on basis of SVM features. AdaRNN-w/E and AdaRNNcomb are based on AdaRNN, which is an RNN wherein sentiment propagations are modelled as semantic compositions. The representations are recursively computed in a bottom-up manner. Dependency types are regarded as features in the adaptive selection of composition functions in AdaRNN-w/E. AdaRNNcomb classifies sentences through an SVM classifier built by combining the vectors obtained from AdaRNN-w/E with bigram and unigram features. AB-LSTM is derived from attention-based bidirectional LSTM. It models the preceding and following contexts using bidirectional LSTM, and then attention is placed on

Bidirectional LSTM, which considers target information. The specific results are shown in the table 1:

Table 1. Experiment Results

Model	Accuracy
SVM	0.627
SVM-dep	0.634
AdaRNN-w/E	0.658
AdaRNNcomb	0.663
AB-LSTM	0.716
Att-CGRU	0.717

It can be known from table 1 that the model proposed in this paper has achieved a better result. The nature of sentiment analysis process is a sentence modeling process, while traditional modeling process of support vector machine, Naive Bayes and other algorithms relies on feature engineering. To compete a feature project by manual work not only consume much time and man power, but also cannot extract effective features. But in case of neural network model, the extraction of features can be automatically completed by nonlinear network. Besides, for aspect level sentiment analysis, aspect is a very important information. By the introduction of attention mechanism, the model can pay more attention to the influence of aspect information and the experiment results our model get better performance.

5. Acknowledgments

This work was supported by the Beijing Science and Technology Planning Program of China (Z171100004717001), Beijing Natural Science Foundation (4162007) and Natural Science Foundation of China (61501008)

6. References

- [1] Nasukawa T and Yi J 2003 *Int. Conf. on Knowledge Capture* (New York: ACM) pp 70–77
- [2] Liu B 2012 *Sentiment analysis and opinion mining* (San Rafael: Morgan & Claypool Publishers) p 1–167.
- [3] Rao D and Ravichandran D 2009 *Conf. of the European Chapter of the Association for Computational Linguistics* (Stroudsburg: ACL) pp 675–682
- [4] Kaji N and Kitsuregawa M 2007 *Conf. on Empirical Methods in Natural Language Processing and Natural Language Learning* (Stroudsburg: ACL) pp 1075–83.
- [5] Yang D and Yang A M 2010 *Application Research of Computers* 27 (10) 3737–39. (In Chinese)

- [6] Pang B, Lee L and Vaithyanathan S 2002 *Conf. on Empirical Methods in Natural Language Processing* (Stroudsburg: ACL) pp 79–86.
- [7] Jiang L, Yu M, Zhou M, Liu X H and Zhao T J 2011 *Annual Meeting of the Association for Computational Linguistics* (Stroudsburg: ACL) pp 151–160.
- [8] Lample G, Ballesteros M, Subramanian S, Kawakami K and Dyer C 2016 *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Stroudsburg: ACL) pp 260-270
- [9] David Golub and Xiaodong He 2016 Character-level question answering with attention *Preprint arXiv/1604.00727*
- [10] Socher R, Pennjngton J, Huang E H, Andrew Y N and Manning C D 2011 *Conf. on Empirical Methods in Natural Language Processing* (Stroudsburg: ACL) pp 151-161.
- [11] Jurgovsky J, Granitzer M 2015 *Int. Conf. on Applications of Natural Language to Information Systems* (Berlin: Springer) pp 160-166.
- [12] Wang Y Q, Huang M L, Zhao L and Zhu X Y 2016 *Conf. on Empirical Methods in Natural Language Processing* (Stroudsburg: ACL) pp 606-615.
- [13] Unanue I J, Borzeshi E Z and Piccardia M 2017 *Journal of Biomedical Informatics* 76 102-109
- [14] Wu S, Zhang D, Zhang Z, Yang N, Li M and Zhou M 2018 *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26(11) 2132-41
- [15] Jozefowicz R, Zaremba W and Sutskever L 2015 *Int. Conf. on Machine Learning*, (New York: ACM) pp 2342-50
- [16] Dong L, Wei F, Tan C, Tang D, Zhou M and Xu K 2014 *Annual Meeting of the Association for Computational Linguistics* (Stroudsburg: ACL) pp 49–54.
- [17] Mnih V, Heess N, Graves A and Kavukcuoglu K 2014 *In Advances in Neural Information Processing Systems* vol 2 (New York: Curran Associates) pp 2204–12
- [18] Liu J, Wang G, Duan LY, Abdiyeva K and Kot AC 2018 *IEEE Trans Image Process* 27(4)1586-99
- [19] Chaplot D, Sathyendra K, Pasumarthi P K, Rajagopal D and Salakhutdinov R 2018 *The AAAI Conf. on Artificial Intelligence* (Menlo Park: AAAI) pp2819-26
- [20] Pennington J, Socher R and Manning C D 2014 *Conf. on Empirical Methods in Natural Language Processing* (Stroudsburg: ACL) pp 1532–43.
- [21] Yang M, Tu W, Wang J, Xu F and Chen X 2017 *The AAAI Conf. on Artificial Intelligence* (Menlo Park: AAAI) pp 5013-14