Word-level sentiment analysis with reinforcement learning

Ruiqi Chen¹*, Yanquan Zhou², Liuje Zhang³ and Xiuyu Duan⁴
¹-⁴School of Computer, Beijing University of Posts and Telecommunications, Beijing, China

*Corresponding author e-mail: chenruiqi@bupt.edu.cn

Abstract. Reinforcement learning (RL) imitates how human perceive the word and acquire knowledge. We manage to propose a RL method to realize text sentiment analysis. We describe a new framework named Word-level Sentiment LSTM (WS-LSTM), which means we use such framework to get sentiment tendency for each word in a sentence. We suppose Positive, Neural and Negative as actions and establish three different LSTM tunnels for each action. When we choose an action for a word, relative LSTM tunnel will be chosen to handle the input. After traversing a whole sentence, we get both word-level sentiment sequence and a sentence-level representation. Such representation is used in classification and we success in getting a sentence-level sentiment analysis. Results show that our method can get sentiment for each word in a specific task. As to the word-level analysis, results from several datasets show that our method plays an acceptable job.

1. Introduction

Sentiment analysis is a significant branch under natural language processing (NLP), first used by Nasukawa and Yi, 2003 [1]. Text sentiment analysis is a computational study of sentiments and, further, opinion. It can provide useful information when we need to make a decision [2].

Researchers manage to use RNN in NLP. Lai et al. construct a mixture neural network with RNN and CNN for text classification [3]. Long Short-Term Memory (LSTM) is an available improvement from Recurrent Neural Networks (RNN). With special structure which is named gate, LSTM is more capable to deal with context information [4]. Sundermeyer et al. propose LSTM with language model [5]. Rao et al. use LSTM with word embeddings to classify text [6]. Wang et al. propose a LSTM model based on attention for aspect-level sentiment analysis [7].

Reinforcement learning is an important branch under machine learning, differing from both supervised learning and unsupervised learning [8]. There are two popular algorithms: Deep Q-Network (DQN) and Actor-Critic Algorithms (AC). Mnih et al. present a deep neural network with the base of Q-learning and establish DQN [9]. DQN has been a popular algorithms with a large number of papers in recent years. Moreover, AC algorithms has two parts, actor and critic, and plays a better job in continuous task [10, 11]. Additionally, some RL algorithms are applied to solve classification task. Wiering et al. models classification task as a sequential problems and propose an actor-critic learning automaton algorithms [12]. Zhang et al. design two LSTM models and propose RL method to discover task-relevant structures.

We design a RL method with LSTM model to build a novel neural network. We use such network to analysis word-level sentiment. The actions are defined as negative, neutral and positive. Actions represent tendency of each word in a sentence. During training, we use a revised ε-greedy which
considers experiential knowledge from existing sentiment dictionaries. We propose three different LSTM cells as tunnels to evaluate the actions. On other word, after choosing an action as actor, we will choose the relative tunnel as critic. The state for agent is represented by the hidden state from LSTM. The tunnels share the same hidden state rather than parameters in memory cell. The output from LSTM cell is used as the structure representation for sentence. As to reward, we calculate cross entropy between predicted results with manual label. Additionally, we include a hyper-parameter to encourage utilizing the experiential knowledge. The reward is delayed so that we can use policy gradient to update model [13].

2. Model

We design a series of actions in analyzing text sentiment in order to imitate human cognitive process as shown in Figure 1. The network is separated into two parts: one is proposed to realize RL algorithms, which we call RL-Network (RLNet); while the other is proposed to realize classification, which we call Classification-Network (CNet). They will be described detailly in follow paper.

![Figure 1. WS-LSTM model.](image)

The RLNet is based on Actor-Critic algorithms. As to the Actor, we choose a sentiment for each word as action. As to the Critic, the action is graded and the stated is changed.

In actor part, when choosing an action for a word, we use a revised ε-greedy algorithm. We managed to build a positive-word dictionary and a negative-word dictionary. Word out of both dictionaries are regarded as neural word. During training, we choose sentiment of the word according to dictionaries with probability 1-ε. By contrast, we arbitrarily choose an action with probability ε. The final policy we need is represented as:

$$\pi(a|s_t, x_t; \theta)$$

where $s_t$ denotes the states at $t^{th}$ timestep, $x_t$ denotes the input at $t^{th}$ timestep, and $\theta$ denotes the parameter of RLNet. Generally, $t^{th}$ timestep refers to the $t^{th}$ word of the sentence.

In critic part, we build three tunnels [14] [Critic_{neg}, Critic_{neu}, Critic_{pos}], where the subscripts refer to {negative, neutral and positive} (similarly hereinafter). The tunnels has same structure, as well as sharing the hidden state which is relevant to agent state. However, they do not share memory cell. With different actions, we will choose the relative tunnel as follow:

$$C_{neg}, h_t = \phi_{neg}(C_{neg}, h_t-1, x_t) \text{ if } a_t = neg$$
$$C_{neu}, h_t = \phi_{neu}(C_{neu}, h_t-1, x_t) \text{ if } a_t = neu$$
$$C_{pos}, h_t = \phi_{pos}(C_{pos}, h_t-1, x_t) \text{ if } a_t = pos$$
where $\varphi$ denotes the function of LSTM structure, $C'$ denotes the previous memory cell, $h_{z-1}$ denotes the previous hidden state, $x_t$ denotes the input from $t^{th}$ word. The last output will be used as the structure representation of the whole sentence.

The representation will be used in analyzing sentence-level sentiment through CNet. The CNet is based on fully-connected layer with dropout and softmax.

### 3. Optimization function

Our model can acquire word-level sentiment as well as sentence-level sentiment result, which can be calculated by cross entropy as loss function:

$$H(p, q) = \sum_x p(x) \log q(x)$$  \hspace{1cm} (3)

where $p(x)$ denotes the predicted result, $q(x)$ denotes the actual label. The loss function is applied to train CNet and Critic part in RLNet.

In order to better use the experiential knowledge, we tend to choose the same sentiment with dictionary when choosing an action. Thus, we define the reward in reinforcement learning as below:

$$R = H + \alpha \left( \frac{L'}{L} \right)$$  \hspace{1cm} (4)

where $H$ is from (3), $\alpha$ is a hyper-parameter for balance, $L'$ denotes the number of word sentiment corresponding to dictionaries and $L$ denotes the length of the sentence.

As to the Actor part, we use the policy gradients to optimize the model:

$$\nabla J(\theta) = E_{\pi}[R_t \nabla \log(\pi(a_t|s_t, \theta))]$$  \hspace{1cm} (5)

As we use the complete return from time to a whole sentence, we apply REINFORCE and eligibility vector. So the (5) is transformed into:

$$\nabla J(\theta) = E_{\pi}[R_t \nabla \log(\pi(a|s, \theta))]$$  \hspace{1cm} (6)

### 4. Experiment

#### 4.1 Datasets and parameter

We evaluated our models with datasets as below:

- **MR:** Movie review. This dataset contains 5331 positive and 5331 negative reviews (Pang and Lee 2005).
- **AF:** Amazon food review. We choose reviews with 5 point grade as positive data and reviews with 1 point grade as negative data.
- **AM:** Amazon mobile review. We choose the data the same as AF.
- **Word Vectors:** Download from (nlp.stanford.edu).

The datasets are separated into {train, test, dev} with ratio 6:3:1. The dropout possibility is set to 0.5. The $\epsilon$-greedy is set to 0.5. The epoch is set to 3 with 10 as mini-batch. The learning rate is 0.001.

#### 4.2 Word-level results

Our model manages to get word-level sentiment after traversing the whole sentence. Examples about word *feel-good* and *unendurable* are listed in Table 1.

Our model succeeds in analyzing the words out of sentiment dictionary. Take *feel-good* as examples. The word refers to positive sentiment but is not mentioned in dictionary. However, in the test dataset. This word appears six times. Four of them are analyzed as positive, while the other two are analyzed as neutral.

In general, there are 58 thousand words in the test dataset. We get the sentiment for each word with our model and compared with the dictionary. The results are show in Table 2. The label *Dictionary+Model* refers to the words which has the same results between dictionaries and our model.
Table 1. Word-sentiment examples

| sentiment | sentence |
|-----------|----------|
| neg       | offers a guilt-free trip into feel-good(NEU) territory. |
| neg       | a feel-good(POS) movie that doesn't give you enough to feel good(NEU) about. |
| pos       | a feel-good(POS) picture in the best sense of the term. |
| pos       | would be an unendurable(NEG) viewing experience for this ultra-provincial(NEU) new… |
| pos       | as crimes go, writer-director michael kalesniko's how to kill your neighbor's dog is slight(NEU) but unendurable(NEG). |

Remark: the label in brackets refers to the sentiment of the word. \{neg, neu, pos\} refers to \{negative, neural, positive\}. Words with no brackets are analyzed as neural sentiment.

Table 2. Word-level results for test dataset

| Sentiment | According              | Count |
|-----------|------------------------|-------|
| neg       | Dictionary             | 2164  |
| neg       | Model                  | 793   |
| neg       | Dictionary+Model       | 164   |
| pos       | Dictionary             | 4655  |
| pos       | Model                  | 879   |
| pos       | Dictionary+Model       | 273   |

4.3 Sentence-level results

The sentence-level classification results is listed in Table 3. The results show that our model does not play better than ID-LSTM in MR dataset, but better than the baselines. All models play a good game in AM and AF datasets. Nevertheless, it proves that our model does not decrease the sentence-level accuracy when analyzing word-level sentiment.

Table 3. Sentence-level sentiment results

| Dataset | MR     | AM     | AF    |
|---------|--------|--------|-------|
| CNN     | 73.8%  | 90.2%  | 94.3% |
| LSTM    | 74.5%  | 96.7%  | 92.2% |
| RCNN    | 76.1%  | 95.2%  | 94.7% |
| ID-LSTM | 81.6%* | --     | --    |
| WS-LSTM | 78.9%  | 93.5%  | 95.1% |

Remark: the result of ID-LSTM is from (Zhang, Huang and Zhao 2017)

Note that our model get the best result with epoch set to 3 and using the 300-dimensional word vectors. Experiments prove that the parameters play important role in training.

5. Discussion and conclusion

We manage to modify the human emotional fluctuation when reading a text. Consequently, we apply reinforcement learning to learn how people do such job. We creatively design three sentiments as actions and use a hidden state to represent agent condition. And we success in combine text sentiment analysis with reinforcement learning.

Our model can get word-level sentiment sequence with a relatively good result through reinforcement learning. However, some words are not judged perfectly. The structure representation for whole sentence is applied to get the sentence-level sentiment and the result is relatively better than the baselines.

It is a hard job to apply reinforcement learning to text classification task. In the future work, we will try to enhance the accuracy for our model. Possible jobs are to modify the gradients and the reward function, or to use tree structures.
References

[1] Nasukawa T, Yi J. Sentiment analysis: capturing favorability using natural language processing[C]// International Conference on Knowledge Capture. DBLP, 2003:70-77.

W. Strunk Jr., E.B. White, The Elements of Style, third ed., Macmillan, New York, 1979.

[2] Liu B. Sentiment Analysis and Opinion Mining[C]// Synthesis Lectures on Human Language Technologies. Morgan & Claypool, 2012:152-153.

[3] Lai S, Xu L, Liu K, et al. Recurrent Convolutional Neural Networks for Text Classification[C]// AAAI. 2015: 2267-2273.

[4] Greff K, Srivastava R K, Koutník J, et al. LSTM: A Search Space Odyssey[J]. IEEE Transactions on Neural Networks & Learning Systems, 2017, 28(10):2222-2232.

[5] Sundermeyer M, Schlüter R, Ney H. LSTM Neural Networks for Language Modeling[C]// Interspeech. 2012:601-608.

[6] Rao A, Spasojevic N. Actionable and Political Text Classification using Word Embeddings and LSTM[J]. 2016.

[7] Wang Y, Huang M, Zhu X, et al. Attention-based LSTM for Aspect-level Sentiment Classification[C]// Conference on Empirical Methods in Natural Language Processing. 2016:606-615.

[8] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[J]. IEEE Transactions on Neural Networks, 2005, 16(1):285-286.

[9] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.

[10] Konda, Vijaymohan. Actor-critic algorithms[J]. Siam Journal on Control and Optimization, 2003, 42(4):1143-1166.

[11] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C]// International Conference on International Conference on Machine Learning. JMLR.org, 2014:387-395.

[12] Wiering M A, Hasselt H V, Pietersma A D, et al. Reinforcement learning algorithms for solving classification problems[J]. 2011:91-96.

[13] Sutton R S. Policy Gradient Methods for Reinforcement Learning with Function Approximation[J]. Submitted to Advances in Neural Information Processing Systems, 1999, 12:1057-1063.

[14] Andre M D L R S, Rieger L, Hannemose M, et al. Tunnel Effect in CNNs: Image Reconstruction From Max Switch Locations[J]. IEEE Signal Processing Letters, 2017, 24(3):254-258.

[15] Tang D, Qin B, Liu T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification[C]// Conference on Empirical Methods in Natural Language Processing. 2015:1422-1432.

[16] Li Y. Deep Reinforcement Learning: An Overview[J]. 2017.

[17] Liu B. Sentiment Analysis and Opinion Mining[C]// Morgan & Claypool, 2012:167.

[18] Pang B, Lee L. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales[C]// Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005:115-124.