Japanese Sentiment Classification using a Tree-Structured Long Short-Term Memory with Attention

Ryosuke Miyazaki*
Graduate School of System Design
Tokyo Metropolitan University
tmcit.miyazaki@gmail.com

Mamoru Komachi
Graduate School of System Design
Tokyo Metropolitan University
komachi@tmu.ac.jp

Abstract

Previous approaches to training syntax-based sentiment classification models required phrase-level annotated corpora, which are not readily available in many languages other than English. Thus, we propose the use of tree-structured Long Short-Term Memory with an attention mechanism that pays attention to each subtree of a parse tree. Experimental results indicate that our model achieves state-of-the-art performance for a Japanese sentiment classification task.

1 Introduction

Traditional approaches for sentiment classification rely on simple lexical features, such as a bag-of-words, that are ineffective for many sentiment classification tasks (Pang et al., 2002). For example, the sentence "Insecticides kill pests." contains both *kill* and *pests*, indicating negative polarity. But, the overall expression is still deemed positive.

To address this problem of polarity shift, Nakagawa et al. (2010) presented a dependency-tree-based approach for the sentiment classification of a sentence. Their method assigns sentiment polarity to each subtree as a hidden variable that is not observable in the training data. The polarity of the overall sentence is then classified by a tree-conditional random field (Tree-CRF), marginalizing over the hidden variables representing the polarities of the respective subtrees. In this manner, the model can handle polarity-shifting operations such as negation. However, this method suffers from feature sparseness because almost all features are combination features.

To overcome the data sparseness problem, deep-neural-network-based methods have attracted much attention because of their ability to use dense feature representations (Socher et al., 2011; Socher et al., 2013; Kim, 2014; Kalchbrenner et al., 2014; Tai et al., 2015; Zhang and Komachi, 2015). In particular, tree-structured approaches called recursive neural networks (RvNNs) have been shown to perform well in sentiment classification tasks (Socher et al., 2011; Socher et al., 2013; Kim, 2014; Tai et al., 2015). Whereas Tree-CRF employs sparse and binary feature representations, RvNNs avoid feature sparseness by learning dense and continuous feature representations. However, annotation for each phrase is crucial for learning RvNN models, but there is no phrase-level annotated corpus in any language other than English.

We therefore propose an RvNN model with an attention mechanism and augment the training example with polar dictionaries to compensate for the lack of phrase-level annotation. Although Kokkinos and Potamianos (2017) also provided an attention mechanism for phrase-level annotated corpus, our model performs well on a sentence-level annotated corpus through the introduction of polar dictionaries.

The main contributions of this work are as follows.

- We show that RvNN models can be learned from a sentence-level polarity-tagged Japanese corpus using an attention mechanism and polar dictionaries.
- We achieve the state-of-the-art performance in a Japanese sentiment classification task.

*Now at Yahoo Japan Corporation.

32nd Pacific Asia Conference on Language, Information and Computation
Hong Kong, 1-3 December 2018

- We have released our code on GitHub.[1]

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 describes our proposed method using Tree-LSTM with an attention mechanism and polar dictionaries. Section 4 presents the experimental results from Japanese and English sentiment datasets, and Section 5 discusses the advantages and disadvantages of the proposed method. Finally, Section 6 concludes our work.

2 Related Work

This section describes related work on Japanese sentiment classification, RvNNs, and attentional models.

2.1 Japanese Sentiment Analysis

Nakagawa et al. (2010) proposed a dependency-based polarity classifier. Their model infers polarity from the composed nodes of a dependency tree using Tree-CRF. Each subtree is represented as a hidden variable in consideration of interactions between the hidden variables. A polar dictionary is used as the initial variable, and a polarity-reversing word dictionary is used to capture whether the constructed phrase polarity is reversed or not. Our model uses only a polar dictionary and attempts to learn polarity shifting via RvNNs.

Zhang and Komachi (2015) adopted a stacked denoising auto-encoder. Their model treats an input sentence as an average vector of their word vectors, which is then fed into a stacked denoising auto-encoder. Although this model omits syntactic information, it achieves high performance through its generalization ability. However, it is not straightforward to employ polar dictionaries in their model.

2.2 Recursive Neural Networks

There are various RvNN models for sentiment classification (Socher et al., 2011; Socher et al., 2012; Socher et al., 2013; Qian et al., 2015; Tai et al., 2015; Zhu and Sobhani, 2015). All of these models attempt to capture sentence representation in a bottom-up fashion in accordance with a parse tree. In this way, sentence representation can be calculated by learning compositional functions for each phrase.

Several studies have focused on using a compositional function to improve compositionality (Socher et al., 2012; Socher et al., 2013; Qian et al., 2015; Tai et al., 2015). Socher et al. (2012) parameterized each word as a matrix–vector combination to denote modification and representation. Socher et al. (2013) used a bilinear function enacted by tensor slicing for composition in place of a large matrix–vector parameterization. Qian et al. (2015) incorporated a constituent label of each phrase as feature embedding to take account of the different compositionality based on parent or children label combinations. Tai et al. (2015) proposed a more robust model than those discussed above in which long short-term memory (LSTM) units were applied to a RvNN, as mentioned in Section 3.1. Thanks to the application of LSTM, this model can learn increased numbers of parameters appropriately, unlike the other models.

2.3 Attentional Models

Based on psychological studies, the human ability of intuition of attention (Rensink, 2000) has been introduced into many computer science fields. The main function of this ability is deciding which part of an input needs to be focused on.

In natural language processing (NLP), the attention mechanism is utilized for many tasks, including neural machine translation (Bahdanau et al., 2015; Luong et al., 2015; Eriguchi et al., 2016), neural summarization (Hermann et al., 2015; Rush et al., 2015; Vinyals et al., 2015), representation learning (Ling et al., 2015), and image captioning (Xu et al., 2015).

These approaches incorporate consideration as to how each source word or region contributes to the generation of a target word. Unlike most of the above-mentioned NLP tasks for generating word sequences, our model derives attention information for sentiment classification. Eriguchi et al. (2016) proposed an attentional model focusing on phrase structure; our model omits the word-level recurrent LSTM layer from their model. Yang et al. (2016) presented a hierarchical attention network for document classification, incorporating sentence- and word-level attention mechanisms. Our task ad-

[1] See https://github.com/tmu-nlp/AttnTreeLSTM4SentimentClassification

dresses sentence-level sentiment classification, so that our model employs a phrase-level (constituent) attention network rather than sentence-level information in a document.

Probably the most related work to our study is (Kokkinos and Potamianos, 2017) and (Zou et al., 2018). Kokkinos and Potamianos (2017) built a similar model to ours, but they did not use the hidden state of a RvNN in their final softmax, and they also did not emphasize the use of polar dictionaries. Zou et al. (2018) proposed a lexicon-based supervised attention model to take advantage of a sentiment lexicon. They injected type-level lexical information into an additional attention network, whereas we injected token-level lexical information into a single RvNN model as a phrase-level annotation.

3 Attentional Tree-LSTM

3.1 Tree-Structured LSTM

Various RvNN models for handling sentence representation considering syntactic structure have been studied (Socher et al., 2011; Socher et al., 2012; Socher et al., 2013; Qian et al., 2015; Tai et al., 2015; Zhu and Sobhani, 2015). RvNNs construct a sentence representation from their phrase representations by applying a composition function. Phrase representations can be calculated by recursively adopting composition functions. Binarizations of parse trees are often used to simplify the composition function. In a parse tree, the root node, non-terminal node, and terminal node represent sentence, phrase, and word representations, respectively.

The ith non-terminal node representation h_i is calculated by using the composition function g as

$$h_i = f(g(h_i^l, h_i^r)), \quad (1)$$

$$g(h_i^l, h_i^r) = W \begin{bmatrix} h_i^l \\ h_i^r \end{bmatrix} + b, \quad (2)$$

where the matrix $W \in R^{d \times 2d}$ and the bias $b \in R^d$ are the parameters to be learned, $h_i^l, h_i^r \in R^d$ are d-dimensional children vectors of node h_i, and the resulting vector h_i is another d-dimensional vector. The hyperbolic tangent is usually employed as the activation function f. These RvNN models are essentially identical to recurrent neural models in that they are not able to retain a long history.

Tai et al. (2015) addressed this problem by introducing LSTM (Hochreiter and Schmidhuber, 1997) to make RvNN less prone to the exploding/vanishing gradient problem. In this paper, we use the Binary Tree-LSTM proposed by Tai et al. (2015) as an example of a tree-structured LSTM. The Binary Tree-LSTM composes children vectors using the following equations:

$$i_j = \sigma\left(U^{(i)} \begin{bmatrix} h_j^l \\ h_j^r \end{bmatrix} + b^{(i)}\right), \quad (3)$$

$$f_{jl} = \sigma\left(U^{(fl)} h_j^r + b^{(fl)}\right), \quad (4)$$

$$f_{jr} = \sigma\left(U^{(fr)} h_j^l + b^{(fr)}\right), \quad (5)$$

$$o_j = \sigma\left(U^{(o)} \begin{bmatrix} h_j^l \\ h_j^r \end{bmatrix} + b^{(o)}\right), \quad (6)$$

$$u_j = \tanh\left(U^{(u)} \begin{bmatrix} h_j^l \\ h_j^r \end{bmatrix} + b^{(u)}\right), \quad (7)$$

$$c_j = i_j \odot u_j + f_{jl} \odot c_{jl} + f_{jr} \odot c_{jr}, \quad (8)$$

$$h_j = o_j \odot \tanh(c_j), \quad (9)$$

where the matrices $U \in R^{d \times 2d}$ (except for U^{fl} and U^{fr}, for which $U \in R^{d \times d}$) and the biases $b \in R^d$ are the parameters to learn. The memory state c is controlled by i, f, and o (called the input gate, forget gate, and output gate, respectively), to hold important information for the entire network. Each gate selectively activates to play a specific role (i.e., the input, forget, and output gates control u_j, c_{jl}, and c_{jr}, respectively, and $\tanh(c_j)$ is based on which elements should be input to the next state c_j, forgotten from the previous state c_{jr} and c_{jl}, or output as a hidden representation h_j, respectively).

Note that the forget gate f_{jl} for the left child state c_{jl} only takes the right child's hidden representation h_j^r and vice versa, as described by Tai et al. (2015).

3.2 Softmax Classifier with Attention

We use a softmax classifier to predict the sentiment label $\hat{y}_j$ at any node j for which a label is to be predicted. Given the jth hidden representation h_j as an input, the classifier predicts $\hat{y}_j$:

$$\hat{y}_j = \arg\max_y \hat{p}_\theta(y|h_j), \quad (10)$$

$$\hat{p}_\theta(y|h_j) = \mathrm{softmax}\left(W^{(s)} h_j + b^{(s)}\right), \quad (11)$$

where $W^{(s)} \in R^{d^l \times d}$ and $b^{(s)} \in R^n$ are the parameter matrix and bias vector for the classifier, respectively, and d^l is the number of labels. The softmax yields a label distribution $y \in R^{d^l}$, following which the classifier chooses the best label corresponding to the highest element among the y.

However, owing to the lack of phrase-level annotation, sentence representation may be inaccurate because it may fail to propagate errors from the root of the tree to the terminals and pre-terminals in a long sentence. We propose an attention mechanism to address this problem. This so-called classifier with attention takes an attention vector representation a_j in addition to a hidden representation h_j as inputs:

$$\hat{p}_\theta(y|h_j) = \mathrm{softmax}\left(W^{(s')} \begin{bmatrix} a_j \\ h_j \end{bmatrix} + b^{(a)}\right), \quad (12)$$

$$a_j = \sum_i a_{ji} \odot h_i, \quad (13)$$

$$a_{ji} = \frac{g(h_i, h_j)}{\sum_{i'} g(h_{i'}, h_j)}, \quad (14)$$

$$g(h_i, h_j) = \exp\left(W^{(a2)} \tanh\left(W^{(a1)} \begin{bmatrix} h_i \\ h_j \end{bmatrix}\right)\right), \quad (15)$$

where $W^{(s')} \in R^{d^l \times 2d}, W^{(a1)} \in R^{d^a \times d}$, and $W^{(a2)} \in R^{1 \times d^a}$ are the parameter matrices. In Eq. 15, the biases for both $W^{(a1)}$ and $W^{(a2)}$ are omitted for simplicity. The attention vector a_j represents how much the classifier pays attention to the children nodes of the target node. The scalar values a_{ji} for each node are used to determine the attention vector.

Figure 1 represents the softmax classifier with attention. Kokkinos and Potamianos (2017) also investigated an attentional model for RvNNs. But, their model only feeds the attention vector into the softmax classifier, whereas our method inputs both the attention vector and RvNN vector, as illustrated in Figure 1.

3.3 Distant Supervision with Polar Dictionaries

Unlike the Stanford Sentiment Treebank, which is annotated with phrase-level polarity, other multilingual datasets contain only sentence-level annotation. As shown in Section 4, sentiment classification without a phrase-level annotated corpus will not learn sentence representations in an appropriate manner. Although a phrase-level-polarity-tagged corpus is difficult to obtain in many languages, polar dictionaries are easy to compile (semi-)automatically. Therefore, we opt for the use of polar dictionaries as an alternative source of sentiment information.

We utilize the same polar dictionaries for short phrases and words as used in Nakagawa et al. (2010). The phrase in the training sets that matches an entry in the polar dictionaries is annotated with the corresponding polarity. The key difference from Nakagawa et al. (2010) is that we use polar dictionaries as a hard label in a manner similar to distant supervision (Mintz et al., 2009). In contrast, the previous work used polar dictionaries as a soft label for an initial hidden variable in Tree-CRF. Teng et al. (2016) also incorporated sentiment lexicons into an recurrent neural network model. Their method predicts weights for each sentiment score of subjective words to predict a sentence label. Our method uses polar dictionaries only during the training step, whereas the method by Teng et al. (2016) needs polar dictionaries for both training and decoding.

3.4 Learning

The cost function is a cross-entropy error function between the true class label distribution, t (i.e., one hot distribution for the correct label) and the predicted label distribution, $\hat{y}$, at each labeled node:

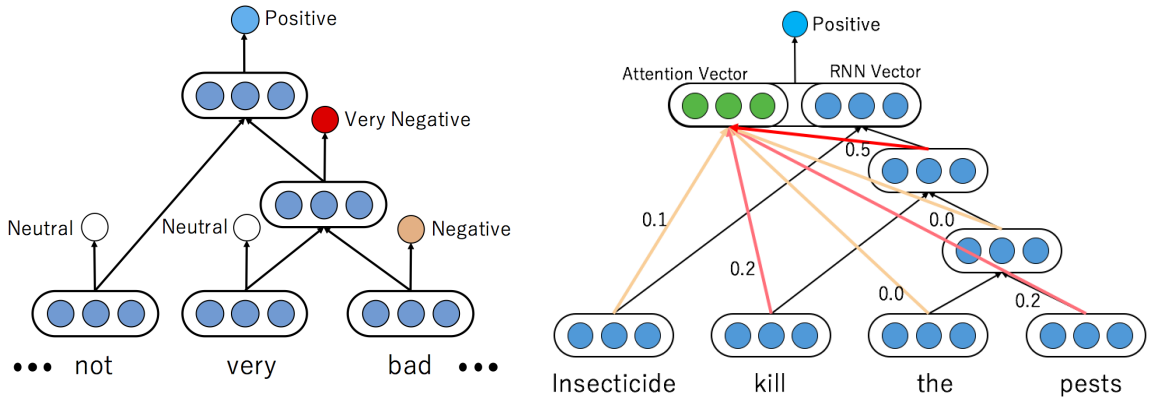
$$J(\theta) = -\sum_{k=1}^{m} t_k \log \hat{y}_k + \frac{\lambda}{2}\|\theta\|_2^2, \quad (16)$$

where m is the number of labeled nodes in the training set[2], and λ denotes an L2 regularization hyperparameter.

4 Experiments on Sentiment Classification

We conducted sentiment classification on a Japanese corpus where phrase-level annotation is unavailable. In addition, we performed sentiment classification on an English corpus where phrase-level annotation is available, but without using phrase-level annotation, to see its effect.

[2]If the dataset contains only sentence-level annotation, m is equal to the size of the dataset.



(a) RvNN (b) RvNN with Attention

Figure 1: Sentiment classification by Tree-LSTM with attention.

4.1 Data

Word embeddings. For Japanese experiments, we obtained pre-trained word representations from word2vec[3] using a skip-gram model (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c). We learned word representations on Japanese Wikipedia's dump data (2014.11) segmented by KyTea (version-0.4.7) (Neubig et al., 2011). We used pre-trained GloVe word representations[4] for the English experiments. We fine-tuned both word representations in our experiments.

Parse trees. For Japanese constituency parsing, we used Ckylark (Oda et al., 2015) as of 2016.07[5] with KyTea for word segmentation. For English, we used the automatic syntactic annotation of the Stanford Sentiment Treebank.

Dictionaries. We followed Nakagawa et al. (2010) to create polar dictionaries. We employed a Japanese polar dictionary composed by Kobayashi et al. (2005) and Higashiyama et al. (2008)[6] that contains 5,447 positive and 8,117 negative expressions.[7] We created an English polar lexicon from Wilson et al. (2005) in the same way as Nakagawa et al. (2010). The dictionary contains 2,289 positive and 4,143 negative expressions.

Corpora. We used the NTCIR Japanese opinion corpus (NTCIR-J), which includes 997 positive and 2,400 negative sentences (Seki et al., 2007; Seki et al., 2008). We removed neutral sentences following previous studies. The corpus comprised two NTCIR Japanese opinion corpora, the NTCIR-6 corpus and the NTCIR-7 corpus, as in (Nakagawa et al., 2010). We performed 10-fold cross-validation by randomly splitting each corpus into 10 parts (one for testing, one for development, and the remaining eight for training).[8] For the English experiments, we used the Stanford Sentiment Treebank (Socher et al., 2013). It includes 11,855 sentences. We followed the official training/development/testing split (8,544/1,101/2,210). We used only sentence-level sentiment for our experiment.

4.2 Methods

In the Japanese experiments, we compared our method with seven baselines. In the English experiments, we compared our method with two baselines. All input word vectors, other than those for most frequent sentiment (MFS) and Tree-CRF, were pre-

[3]See https://code.google.com/archive/p/word2vec/

[4]See http://nlp.stanford.edu/data/glove.6B.zip

[5]See https://github.com/odashi/ckylark

[6]See http://www.cl.ecei.tohoku.ac.jp/index.php?OpenResources/JapaneseSentimentPolarityDictionary

[7]Note that these figures are slightly different from Nakagawa et al. (2010). We suspect that the reason why they can use a larger lexicon is that they used an in-house (not publicly available) version of the lexicon.

[8]Nakagawa et al. (2010) did not use development data to train the model, which means our model uses only 86.6% of the instances to train the model compared with theirs.

trained by word2vec in Japanese experiments and by GloVe in English experiments. We implemented our method, LogRes, RvNN, Tree-LSTM, and our reimplementation of Kokkinos and Potamianos (2017) using Chainer (Tokui et al., 2015).

The following methods were used.

MFS. A naïve baseline, as it always selects the most frequent sentiment (which is negative in this case).

LogRes. A linear classifier using logistic regression. The input features are an average of word vectors in a sentence.

CNN The CNN-based sentiment classification (Kim, 2014).[9]

Tree-CRF. A dependency-based tree-structured CRF (Nakagawa et al., 2010). This is the state-of-the-art method among our experimental datasets.[10]

RvNN. The simplest RvNN.

Tree-LSTM. The LSTM-based RvNN (Tai et al., 2015).

Kokkinos and Potamianos (2017) Our reimplementation of Kokkinos and Potamianos (2017). We implemented their TreeGRU model with LSTM instead of GRU.

Tree-LSTM w/ attn, dict. Our proposed method, which classifies polarity using attention and/or polar dictionaries.

4.3 Hyperparameters

The parameters used in both experiments are listed in Table 1. For Japanese experiments, we tuned hyperparameters on each development set of 10-fold cross-validation. For English experiments, we used similar hyperparameters with slight modifications to the Japanese experiments.

4.4 Results

The Japanese experimental results are listed in Table 2. The accuracy of RvNN is much lower than that of the MFS baseline. Moreover, Tree-LSTM, which is an improved RvNN, is still lower than simple LogRes, despite Tree-LSTM achieving state-of-the-art performance on the phrase-annotated Stanford Sentiment Treebank (Tai et al., 2015). The accuracy of CNN and Kokkinos and Potamianos (2017) is 0.803 and 0.807, respectively, which are slightly lower than that of our proposed method without polar dictionaries. In contrast, Tree-LSTM with dictionary achieves comparable results to Tree-CRF. Our Tree-LSTM with attention and polar dictionary obtained the highest accuracy.

The results of the English experiments are listed in Table 3. Similarly to Kokkinos and Potamianos (2017), we observe slight improvement when using attention for Tree-LSTM. However, unlike the Japanese experiments, using a dictionary degrades sentiment classification accuracy. We discuss this in the following section.

Parameter	Value	
	Japanese	English
Word vector size	200	300
Hidden vector size	200	200
Optimizer	AdaDelta	AdaGrad
(Weight decay/learning rate)	0.0001	0.005
Gradient clipping	5	5

Table 1: The hyperparameters.

Method	Accuracy
MFS	0.704
LogRes	0.771
CNN (Kim, 2014)	0.803
Tree-CRF (Nakagawa et al., 2010)	0.826
RvNN	0.517
Reimplementation of Tai et al. (2015)	0.709
Reimplementation of K&P (2017)	0.807
Tree-LSTM w/ attn	0.810
Tree-LSTM w/ dict	0.829
Tree-LSTM w/ attn, dict	**0.844**

Table 2: Accuracy of each method on the Japanese sentiment classification task.

[9] See https://github.com/yoonkim/CNN_sentence

[10] Note that we cannot directly compare our result with a stacked denoising auto-encoder (Zhang and Komachi, 2015) (which achieved slightly higher accuracy in the NTCIR-6 corpus,) because we use a different dataset in our experiment.

Method	Accuracy
(Kim, 2014)	
— CNN	48.0
(Tai et al., 2015)	
— Tree-LSTM	51.0
(Kokkinos and Potamianos, 2017)	
— TreeGRU w/o attn	50.5
— TreeGRU w/ attn	51.0
Tree-LSTM	43.52
Tree-LSTM w/ attn	44.97
Tree-LSTM w/ dict	43.13
Tree-LSTM w/ attn, dict	41.67

Table 3: Accuracy of each method on the English sentiment classification task. Note that our model learns only from sentence-level annotation.

5 Discussion

5.1 Effect of Attention and Dictionary

The results described above indicate that Tree-LSTM models without an attention mechanism fail to learn sentence representations if phrase-level annotation is not available.

However, Tree-LSTM models can learn more accurate sentence representations if the models receive phrase-level information such as that provided by polar dictionaries. For example, in our model, attention information and a polar dictionary are fed into the Tree-LSTM as phrase-level information. Although Tree-LSTM with attention and a polar dictionary outperforms Tree-CRF by 1.8 points, the accuracy of CNN, Tree-LSTM without a polar dictionary, and Kokkinos and Potamianos (2017) are lower than that of Tree-CRF. Tree-LSTM with a polar dictionary performs better than Tree-LSTM with attention, showing that a supervised label for each phrase seems to be important in learning Tree-LSTM models.

Table 3 indicates that although an attention mechanism is effective in both sentence-level and phrase-level annotated corpora, the effect of dictionary information varies across datasets. It is known that the effect of the English lexicon (Wilson et al., 2005) is milder in the review dataset than in other domains (Nakagawa et al., 2010), and we suppose that it introduces noise in the Stanford Sentiment Treebank.

5.2 Examples

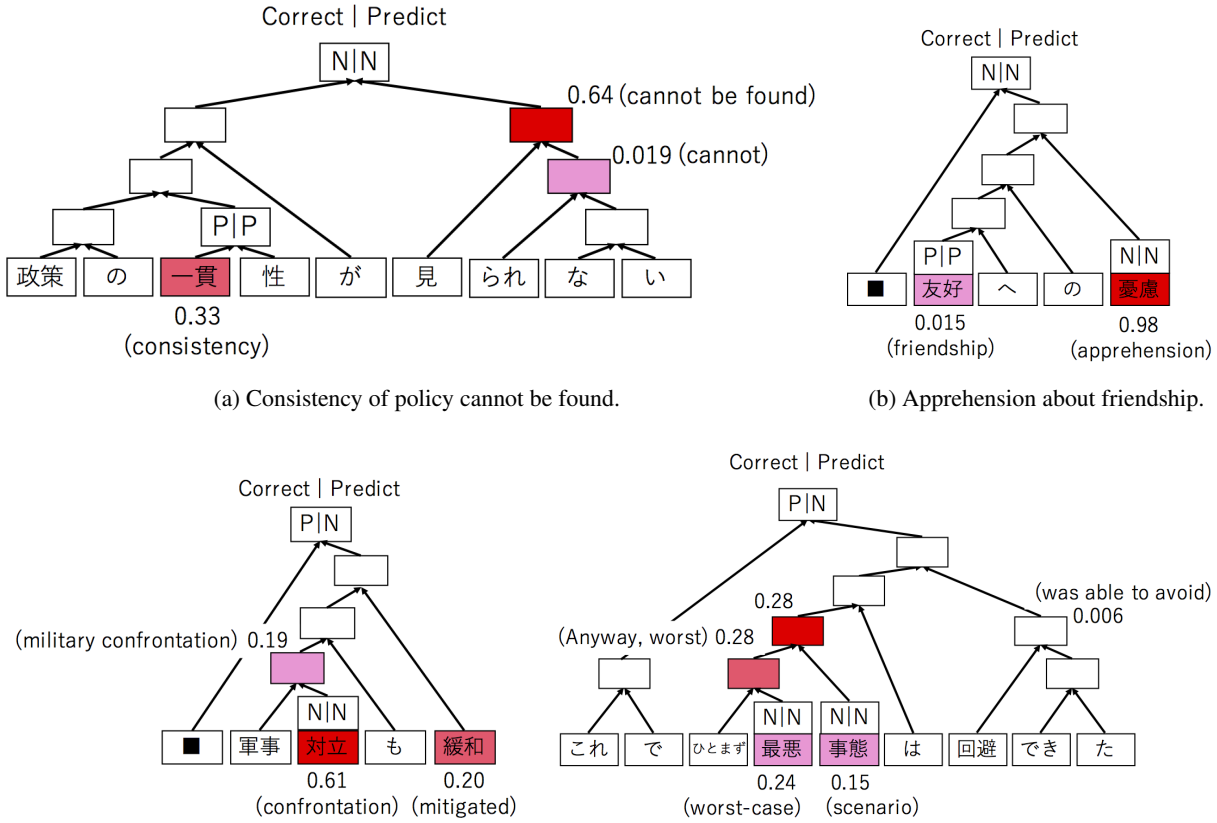
Figures 2a and 2b show correctly classified examples. Figure 2a shows that the model classifies "一貫性 (consistency)" as positive and pays 1/3 attention to it in the final classification step; however, the model correctly classifies the sentence polarity as negative by considering "見られない (cannot be found)" through most of the attention. In Figure 2b, the model correctly classifies both "友好 (friendship)" and "憂慮 (apprehension)," and then classifies the sentence polarity by paying great attention to "憂慮 (apprehension)."

Figures 2c and 2d display an incorrectly classified example. In Figure 2c, the model pays attention to both "対立 (confrontation)" and "緩和 (mitigated)"; however, it fails to predict the correct polarity of the sentence. It seems that the higher attention weight for "対立 (confrontation)" than for "緩和 (mitigated)" has an influence on the sentence prediction. In Figure 2d, the model fails to capture negation as it should pay attention to "回避できた (was able to avoid)." To solve these errors, the composition function should also incorporate an attention mechanism to handle polarity shifting correctly.

6 Conclusion

We have presented a Tree-LSTM-based RvNN using an attention mechanism and a polar dictionary. In this method, each phrase representation is fed into a classifier to predict the polarity of a phrase based on the phrase structures. Lexical items from the polar dictionary are used as supervised labels for each corresponding phrase or word in the same manner as distant supervision. Our experimental results have demonstrated that the proposed method outperforms the previous methods on a Japanese sentiment classification task.

Currently, we have annotated fine-grained phrase-level sentiment tags to a Japanese review corpus. We plan to analyze the effect of phrase-level annotation on Japanese sentiment analysis. In addition, we would like to extend our attentional mechanism to use the Transformer (Vaswani et al., 2017) as proposed in Shen et al. (2018).



(a) Consistency of policy cannot be found.

(b) Apprehension about friendship.

(c) Military confrontation was mitigated.

(d) Anyway, it was able to avoid the worst-case scenario.

Figure 2: Examples of our attentional Tree-LSTM sentiment classification on the test set. The red square indicates a word or phrase to which great attention was paid in the softmax step, and the associated value indicates the attention weight. Root nodes are indicated by the (left) gold and (right) predicted labels ("N" indicates negative, whereas "P" indicates positive). We also show the labels for nodes that match an entry in the polar dictionary.

References

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Natural Language Processing*, page 823–833.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28*, pages 1693–1701.

Masahiko Higashiyama, Kentaro Inui, and Yuji Matsumoto. 2008. Learning sentiment of nouns from selectional preferences of verbs and adjectives. In *Proceedings of the 14th Annual Meeting of the Association for Natural Language Processing*, pages 584–587.

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

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Natural Language Processing*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, and Kenji Tateishi. 2005. Collecting evaluative expres-

sions for opinion extraction. *Journal of Natural Language Processing*, 12(3):203–222.

Filippos Kokkinos and Alexandros Potamianos. 2017. Structural attention neural networks for improved sentiment analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 586–591.

Wang Ling, Lin Chu-Cheng, Yulia Tsvetkov, Silvio Amir, Ramón Fernandez Astudillo, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Not all contexts are created equal: Better word representations with variable attention. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1367–1372.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *Proceedings of Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 746–751.

Mike Mintz, Steven Bills, Rion Snow, and Dan jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. 2010. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786–794.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable Japanese morphological analysis. In *Proceedings of The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Ckylark: A more robust PCFG-LA parser. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 41–45.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Qiao Qian, Bo Tian, Minlie Huang, Yang Liu, Xuan Zhu, and Xiaoyan Zhu. 2015. Learning tag embeddings and tag-specific composition functions in recursive neural network. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1365–1374.

Ronald A. Rensink. 2000. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Yohei Seki, David Kirk Evans, and Lun-Wei Ku. 2007. Overview of opinion analysis pilot task at NTCIR-6. In *Proceedings of the 6th NTCIR Workshop*, pages 265–278.

Yohei Seki, David Kirk Evans, and Lun-Wei Ku. 2008. Overview of multilingual opinion analysis task at NTCIR-7. In *Proceedings of the 7th NTCIR Workshop*, pages 265–278.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2018. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5446–5455.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161.

Richard Socher, Brody Huvaland Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211.

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. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1556–1566.

Zhiyang Teng, Duy-Tin Vo, and Yue Zhang. 2016. Context-sensitive lexicon features for neural sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1629–1638.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems in The 29th Annual Conference on Neural Information Processing Systems*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28*, pages 2692–2700.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the 2005 Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2048–2057.

Zichao Yang, Diyu Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 1480–1489.

Peinan Zhang and Mamoru Komachi. 2015. Japanese sentiment classification with stacked denoising auto-encoder using distributed word representation. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 150–159.

Xiaodan Zhu and Parinaz Sobhani. 2015. Long short-term memory over recursive structures. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1604–1612.

Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2018. A lexicon-based supervised attention model for neural sentiment analysis. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 868–877.