Expanding the Feature Space of Deep Neural Networks for Sentiment Classification

Mate Kovacs and Victor V. Kryssanov

Abstract—**Deep learning has made remarkable advances in many application domains, demonstrating its robustness in various data mining tasks. However, it is often overlooked that external information sources can explicitly be included in deep learning models to expand their feature space for improved performance, especially when large datasets are not available. This paper presents a neural network architecture for multi-class sentiment analysis, incorporating semantic information from a sentiment lexicon. The model was evaluated on a small dataset of Japanese hotel reviews, with results indicating that integrating sentiment polarities into neural networks can increase classification accuracy.**

Index Terms—**Sentiment analysis, sentiment lexicon, deep learning, text mining, customer reviews.**

I. Introduction

In these days, companies collect and store a large amount of customer data in order to enable better business decisions and gain an advantage in the global market. Interacting with the customers and acquiring information about their needs, concerns, and attitudes is strongly related to the success of online buying and selling services. Many e-commerce websites, such as Amazon, Rakuten, Yahoo, TripAdvisor, etc., allow the customers to leave their opinion about the products and services the company offers. Electronic word-of-mouth influences product sales [1], as customers often consider these when making purchasing decisions, and tend to rely on online reviews more than product advertisements [2]. Moreover, these easily obtainable feedbacks are also a valuable source of information for the companies, as reviews are generated without much corporate effort. However, extracting customer intelligence from such user-generated content is a challenging task, as it involves dealing with data from natural language source.

Sentiment analysis (SA) is the process of computationally identifying opinions, sentiments, emotions, attitudes toward entities and their attributes [3]. SA from textual data is one of the most popular research areas of text mining and attracts an increasing attention from the research community and the industry as well. Companies often conduct SA on online reviews and analyze the emotions and attitudes of the customers in order to improve the quality of the services and products they offer. SA can be performed on three different levels [4]. While on the document level, sentiments are analyzed considering the whole document, sentence level sentiment analysis deals with extracting sentiments from individual sentences. Lastly, aspect-based approaches involve identifying people's opinions on target aspects of an entity. SA usually involves the identification of sentiment orientation. This can be defined as a simple binary classification task (positive or negative), or as multi-class classification often involving categories of a higher abstraction (e.g. emotions, attitudes), introducing a more challenging task.

Most of the early work in sentiment analysis utilized polarity dictionaries and opinion lexicons to perform sentiment classification, but machine learning based methods are usually more robust and outperform knowledge-based approaches [5]. Because they eliminate the necessity of manual feature engineering, deep learning algorithms are greatly advantageous in many application domains and are also broadly used for SA. One of the reasons of the success of deep learning in text mining is that modeling language requires learning the dependencies between sequential elements, and neural networks of recurrent type are exceptionally good at learning such temporal dynamics. However, deep learning usually requires a large amount of labeled data to achieve satisfactory performance [6], which takes a considerable amount of time and money to produce, or just simply unfeasible to obtain.

Although hybrid approaches exist that combine knowledge-based approaches with machine learning methods, limited work has been done on explicitly expanding the feature space of deep neural networks by external knowledge. In the present study, linguistic information from a sentiment lexicon is incorporated into a deep learning architecture in order to improve multi-class sentiment classification performance on a small dataset. The categories to be predicted in the dataset are praise, complaint, request, neutral, and no sentiment on the sentence level. These constitute a more complex classification case than the usual binary scheme, because of the higher level of ambiguity and annotation subjectivity. Results obtained from the experiments demonstrate that it is possible to effectively integrate external semantic information into deep neural networks, and improve sentiment classification performance.

The rest of the paper is organized as follows. Section II addresses related literature, and Section III introduces the

Manuscript received May 12, 2019; revised December 9, 2020.

Mate Kovacs is with the Graduate School of Information Science and Engineering, Ritsumeikan University, 525-8577 Kusatsu, Nojihigashi 1-1-1, Japan (e-mail: gr0370hh@ed.ritsumei.ac.jp).

Victor V. Kryssanov is with the Collage of Information Science and Engineering, Ritsumeikan University, 525-8577 Kusatsu, Nojihigashi 1-1-1, Japan (e-mail: kvvictor@is.ritsumei.ac.jp).

doi: 10.18178/ijmlc.2020.10.2.931

data used in this study. Section IV describes the proposed neural network architecture in detail, and Section V presents the experimental settings used for the evaluation of the model. Results obtained are summarized and discussed in Section VI. Finally, conclusions are drawn and future work directions are outlined in Section VII.

II. Related Work

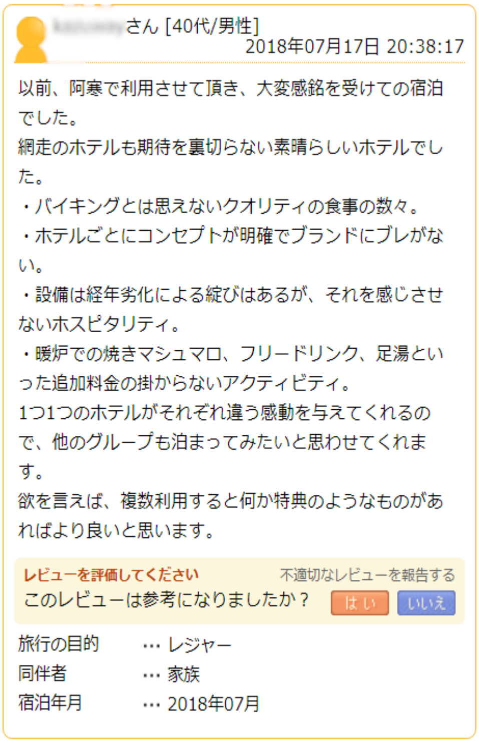
Several neural network models have been proposed for SA. Recurrent neural networks such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) models, and Convolutional Neural Networks (CNN) became very popular in text mining, as these models can encode semantic and syntactic information without parse trees [7]. These models usually make use of word embeddings, *n*-dimensional vector representations of words based on their semantic relationships and context.

Wang *et al.* [8] described a combined architecture of CNN and different RNN models for sentiment classification on short texts. This joint model proved to be an effective combination to learn local features and long-term dependencies at the same time. Huang *et al.* [9] proposed a tree-structured LSTM where POS tags of the words manage the gates of the recurrent network to encode additional syntactic knowledge into the model. Appel *et al.* [10] proposed a hybrid approach for sentence-level sentiment analysis. The system utilizes a sentiment lexicon and SentiWordNet with the combinations of fuzzy sets to predict the polarity of the sentences. Ebert *et al.* [11] argued that linguistic knowledge can be useful in sentiment classification. The authors described a CNN architecture, where word sentiments were added to their embedding vectors, and additional sentence-level features were augmented into the network to improve classification accuracy in a positive-neutral-negative classification case. Teng *et al.* [12] proposed a method for sentence level sentiment classification. The approach relies on the context of the sentiment words and uses an LSTM model to learn sentiment strength to compose a final sentiment value for the sentences. One of the works which use a different classification scheme than the conventional binary (or binary + neutral) categorization is presented by Guggilla *et al.* [13]. Their model combines a CNN model with LSTM for classifying online argument claims as factual or emotional.

Although some of the previous work recognize the necessity of including external information into deep learning models, existing approaches do not consider integrating information from sentiment lexicons in an explicit way, which would allow the network to learn more powerful representations of the sentences.

III. Data

The dataset used in the study is the Tsukuba sentiment-tagged corpus, created by “NLP on the Web” Laboratory of Tsukuba University, and provided by the Rakuten, Inc. (the dataset is distributed by the National Institute of Informatics). The dataset consists of 4309 Japanese sentences tagged with sentiments from Rakuten Travel's hotel reviews (https://travel.rakuten.co.jp). Rakuten Travel is an online hotel reservation service, where the users can also express their thoughts and suggestions for the hotels, in the form of reviews. An example of such a review from is shown in Fig. 1.



[redacted]さん [40代/男性]
2018年07月17日 20:38:17

以前、阿寒で利用させて頂き、大変感銘を受けての宿泊でした。
網走のホテルも期待を裏切らない素晴らしいホテルでした。
・バイキングとは思えないクオリティの食事の数々。
・ホテルごとにコンセプトが明確でブランドにブレがない。
・設備は経年劣化による綻びはあるが、それを感じさせないホスピタリティ。
・暖炉での焼きマシュマロ、フリードリンク、足湯といった追加料金の掛からないアクティビティ。
1つ1つのホテルがそれぞれ違う感動を与えてくれるので、他のグループも泊まってみたいと思わせてくれます。
欲を言えば、複数利用すると何か特典のようなものがあればより良いと思います。

レビューを評価してください
不適切なレビューを報告する
このレビューは参考になりましたか？
はい
いいえ

旅行の目的	… レジャー
同伴者	… 家族
宿泊年月	… 2018年07月

Fig. 1. An example of hotel review from Rakuten travel.

The labeling was done by two annotators, and the annotation scheme was the following:

1) The possible categories are 'praise' 「褒め」, 'complaint' 「苦情」, 'request' 「要求」, 'neutral' 「ニュートラル」, and ‘no sentiment’ 「評価なし」.
2) If there are several sentiments in the same sentence, annotator 1 used plural labels, while annotator 2 used a single label.
3) Sentences that are not given at least one of the labels described above are tagged as 'pending' 「その他/保留」.

Because pending sentences are practically useless, these were removed from the data. This resulted in a final 4219 sentences to use. When annotator 1 used several labels for a given sentence, the label from annotator 2 was considered. When annotator 1 and annotator 2 did not agree on a single label, a random choice was made between them to decide which label is going to be used, in an unbiased manner. Table I. shows the final number of examples for each category in the dataset.

TABLE I: Number of Class-Wise Sentences in the Corpus

praise	complaint	request	neutral	no sentiment
1846	827	201	280	1065

IV. Proposed Architecture

The proposed network architecture is shown in Fig. 2.

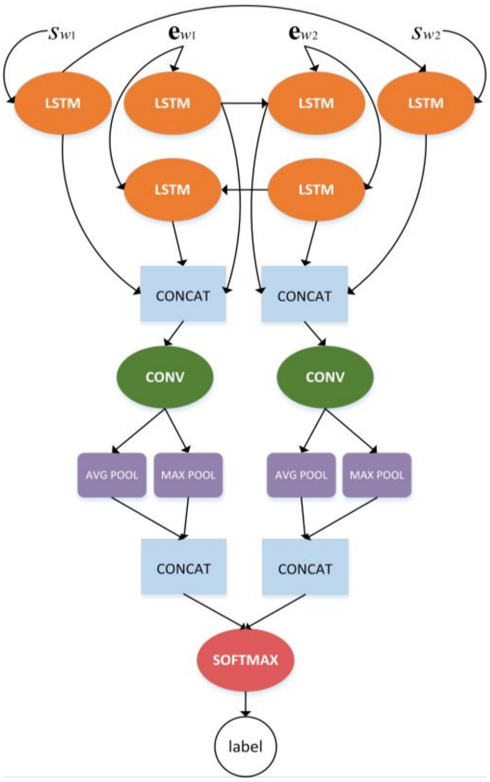


Fig. 2. The proposed deep learning architecture (see main text for notations).

There are two types of inputs for the network; word embedding vectors $\mathbf{e}_w$ and word sentiment vectors s_w. A Japanese sentiment lexicon was used to create the sentiment vectors s_w. The sentiment lexicon created by Takamura *et al.* [14] provides polarity scores for nouns, adjectives, verbs, and adverbs. In order to overcome the problem of manual annotation, the authors applied an automatic approach to extract word polarities with a small number of seed words. As the scores obtained from the lexicon are already on a continuous scale from -1 to 1, these were used as sentiment vectors in the network. For example, the sentiment vector for the noun 発達 (‘development’) is 0.877737, or -0.990751 for the adjective 訝しい (‘doubtful’). Score of -1 indicates that the word is undoubtedly negative, and 1 shows that the word is definitely positive. For the words in the dataset which the sentiment lexicon does not include, the score of 0.0 was used to indicate the neutral nature of these words. To create the embedding vectors $\mathbf{e}_w$, Global Vector (GloVe) [15] is used in this study. In contrast to predictive embedding models (such as word2vec), word vectors are learned by constructing a word-word co-occurrence matrix, then applying matrix factorization to obtain word representations in a lower-dimensional vector space. While predictive methods usually rely on local information (direct context of words), GloVe vectors capture not just the local, but the global information as well. This makes it preferable for many natural language processing tasks, and it is a widely used method for obtaining word vectors for machine learning applications. In order to learn as accurate embeddings as possible, the GloVe model was pre-trained on Japanese Wikipedia dump data (https://dumps.wikimedia.org/jawiki).

To learn sequential relations between the words based on their sentiment orientation, the s_w vectors are used as an input for an LSTM layer. Unlike fully-connected networks, LSTM is a recurrent-type of neural network which excels in processing sequential information. In contrast to conventional recurrent networks, LSTM is capable of learning dependencies on the long-term, and solves the problems of vanishing and exploding gradients. This makes it very popular for time-series prediction and for text processing [4]. The formal representation of an LSTM building block is shown in Fig. 3.

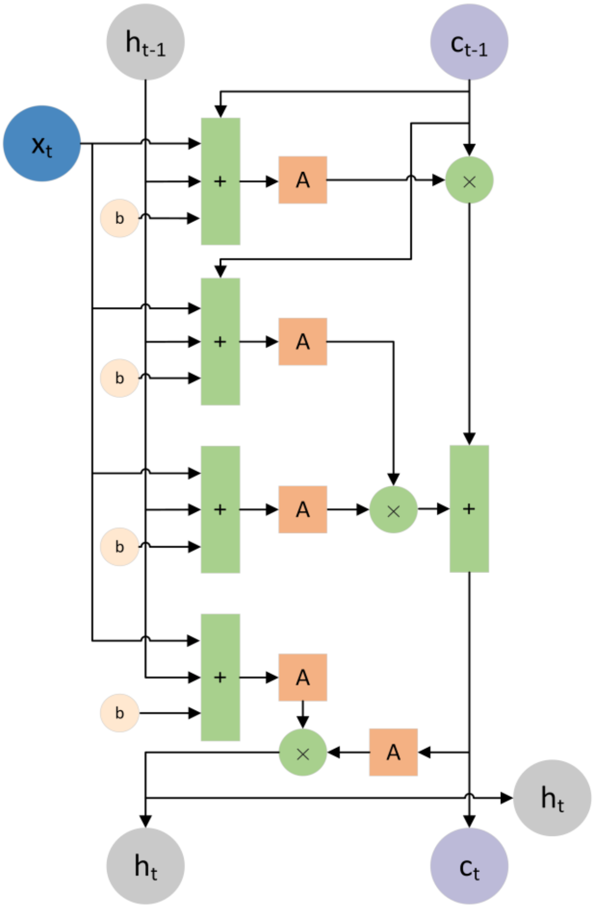


Fig. 3. LSTM building block (see main text for notations).

The inputs for an LSTM block are the current input x_t at a given time t, the output from the previous hidden state h_{t-1}, and c_{t-1} that is the internal memory from the previous block. This integrated memory controls how much information should be kept from the previous states, and also the volume of new memories. The × and + operations are element-wise multiplication and summation respectively. While *A* indicates the activation functions, *b* denotes the biases. The outputs are the hidden state h_t, and the internal memory from the current block c_t. Therefore, the hidden state is depending on the previous state of the recurrent network, not just the current input. This means that LSTM is processing the text in a forward direction, similar way as humans do.

The word embedding vectors $\mathbf{e}_w$ are fed into a Bidirectional LSTM (BiLSTM) [16]. Fig. 4 depicts the architecture of a BiLSTM model for three timesteps. This special kind of recurrent layer is processing data from two directions to obtain an output y_t for timestep *t*. Because of this unique characteristic of BiLSTM type of recurrent networks, these are widely used in Speech Recognition, Named Entity Recogntion, Automatic Translation, Sentiment Analysis, etc. A BiLSTM block consists of two separate hidden LSTM layers, a forward and a backward layer. This practically means that one layer is processing the text from left to right, and the other is from the right to the left. The two directions of LSTM do not have direct connections in the network.

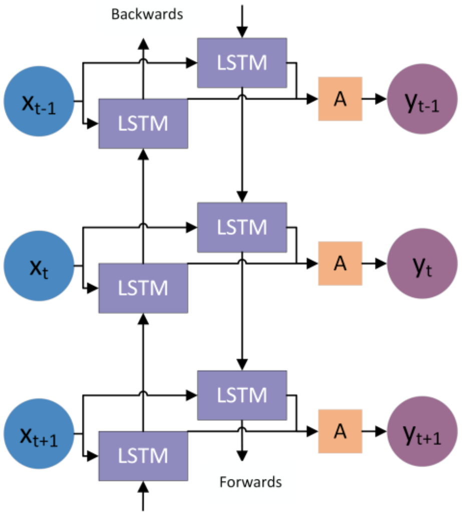


Fig. 4. Bidirectional LSTM architecture unfolded for three timesteps.

These hidden states h_{t-1} and h_{t+1} are then concatenated with the LSTM output of the sentiment vectors (see Fig. 2), to make a joint representation of the sentence from the two layers (CONCAT). The concatenated outputs then used as inputs for a 1-dimensional convolutional layer to allow the network to find local features regardless of their positions in the sentence sequences. In order to decrease the spatial size of the learned representation, max pooling (MAX POOL) and average pooling (AVG POOL) is applied after the convolutional layer. This allows the network to lower the number of features, but retain most of the useful information about the classes. Finally, the concatenated outputs from the pooling layers are fed into a fully-connected layer with a softmax activation function. Softmax is the generalization of the logistic function:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \quad (1)$$

where *x* is the input vector to the output, and $j \in \{1, 2, 3, ..., K\}$ indicates the output unit index. The output of softmax is essentially the ratio of the exponential of an input vector's given unit, and the sum of exponentials of all the units in the input. As the output can be interpreted as probabilities, this makes softmax ideal for multi-class prediction.

V. Experiments

Before feeding the sentences into the network, the dataset was preprocessed. After stop-words were eliminated from the review texts, the sentences were tokenized, and special characters were removed. The performance of the proposed architecture was assessed by comparing it to two other models. First, a vanilla neural network with a single hidden layer was trained on the dataset to get baseline results for comparison. Then, the BiLSTM+CNN model described in Section IV was trained with and without using knowledge from the sentiment lexicon in order to evaluate the influence of incorporating external linguistic information into the network. The number of LSTM neurons learning from the sentiment vectors were set to 32 because of the one-dimensional inputs. To integrate complex dependencies into the word vectors, embeddings of 200 dimensions were used in the network. Thus, the forward and backward BiLSTM layers learning from the embedding vectors had 200 neurons each. To prevent overfitting, dropout was applied on these layers with the probability of 0.3. For the convolutional layer, 64 filters were used with the size set to only 3, in order to capture local features. The network was trained with batch sizes of 128 samples. Due to the small size of the dataset and the unbalanced distribution of class instances, 10-fold cross-validation was performed to train and test the models. Training error was calculated by using categorical cross entropy, to accommodate for the multi-class nature of the classification task:

$$L_i = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} t_{i,c} \log\left(p_{i,c}\right) \quad (2)$$

where $i \in \{1, 2, 3, ..., N\}$ is an observation, and $c \in \{1, 2, 3, ..., C\}$ is a class. $t_{i,c}$ is the target indicator function denoting that *i* belongs to class *c*, and $p_{i,c}$ are the predicted probabilities of *i* belonging to *c*.

The code for the neural network was implemented in Python 3.7, using the Keras library. The tokenization was performed with Janome, a Japanese morphological analyzer. The GloVe vectors was pre-trained, and the network was trained on a workstation with an Intel Xeon E5-1650 v4 CPU, 128GB DDR4 RAM, and an Nvidia 2080ti GPU.

VI. Results and Discussion

All three models were trained for 10 epochs, and testing accuracy was recorded upon each fold of the cross-validation. Final accuracy scores were calculated by averaging these accuracies over all folds. The baseline NN model with one hidden layer resulted in 67.02% classification accuracy on the test set. By using the combined model of BiLSTM and CNN, the results are already significantly better compared to the baseline network. 75.62% accuracy was obtained by this configuration. However, using the polarity scores obtained from the sentiment lexicon with the BiLSTM+CNN yielded the best results of 80.4% accuracy, surpassing the two other models by ~13.38%, and ~4.78% respectively. This suggests that sentiment scores capturing only the polarity of words can help identifying sentiments at higher abstraction levels if assessed sequentially. Because of the increased number of parameters, the network became more prone to overfitting when the sentiment scores were included to the model. For this reason, L2 regularization was added to the convolutional layer to prevent weights from overfitting the training examples.

The class-wise precision, recall, and F1 values for BiLSTM+CNN+sentiment lexicon model are shown in Table II. While classifying 'praise' and 'complaint' sentences is relatively a straightforward task, identifying 'neutral' sentences proved to be more challenging. This can be due to the fact that 'no sentiment' and 'neutral' are differentiated in the annotation scheme. Neutral sentiment label means that there are positive and negative sentiments mixed in the sentence, which makes it "neutral". For example; "夕食ルームサービスは全体として満足ですが、メインの鍋焼味噌カツは味が濃すぎでした" ("I was satisfied with the dinner room service overall, but the hot pot miso pork cutlet's taste was too strong"). This mixed nature of the category and the small number of instances makes this class ambiguous, and harder to predict than the others.

TABLE II: CATEGORY-WISE PERFORMANCE MEASURES FOR THE PROPOSED NETWORK ARCHITECTURE

Performance measure	Precision	Recall	F_1
praise	0.874	0.89	0.882
complaint	0.766	0.773	0.769
request	0.758	0.63	0.688
neutral	0.447	0.426	0.436
no sentiment	0.819	0.814	0.816

Sentences labeled as 'no sentiment' are usually objective statements or facts. The lack of stronger sentiment words and the higher number of instances makes this class simple to learn with the proposed network. Interestingly, despite the small number of training examples, the F1 score of class 'request' is rather high. The reason for this can be that making a request, suggestion, or demanding something makes constraints on the grammatical structure of the sentence and word choice. For instance; "...改善する余地がある" ("... has a place for improvement"), or"...して欲しかった" ("wanted them to...."). This makes this category considerably easier to classify than 'neutral'.

VII. CONCLUSIONS

In this work, semantic information from a sentiment lexicon is incorporated into a deep learning architecture to improve the performance on a multi-class sentiment classification task. Experiments were conducted on the Tsukuba sentiment-tagged corpus, a Japanese dataset annotated on the sentence level. Classification accuracy was increased by ~13.38% compared to a baseline neural network, within ~4.78% can be credited for using external knowledge from the sentiment lexicon. This shows that the proposed network architecture can be applied effectively to classify sentiments of review sentences. Furthermore, results indicate that besides word embeddings, external knowledge sources can potentially enrich the feature set for natural language processing tasks, especially when large datasets are not available. The model used in this study was evaluated on a Japanese dataset, but it would be applicable to any language with a polarity dictionary.

While 10-fold cross-validation was applied to deal with the problem of unbalanced dataset, upsampling techniques could be utilized to learn more accurate representations of the sentences in classes with a smaller number of examples. Future work can expand the current study by increasing the number of external features and applying the proposed network architecture to a sentiment classification task with different categories from those considered in this study. Other line of research could experiment using different word embeddings, or learning the word vectors while the model is trained.

DATA AVAILABILITY

The data used in this study (Tsukuba sentiment-tagged corpus) is available from the National Institute of Informatics, Japan, but restrictions may apply to the availability of the data.

CONFLICT OF INTEREST

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial or non-financial interest in the subject matter or materials discussed in this manuscript.

AUTHOR CONTRIBUTIONS

Mate Kovacs made substantial contributions to conception of the research process and the analysis of data. Both authors participated in the interpretation of data and drafting the article. The code for the neural network model was implemented by Mate Kovacs. While the draft was written by Mate Kovacs, Victor V. Kryssanov revised it critically for important intellectual content. Both authors had approved the final version.

ACKNOWLEDGMENT

The authors acknowledge the use of the Tsukuba sentiment-tagged corpus, part of the Rakuten Data provided for research purposes by the National Institute of Informatics, Japan, in cooperation with Rakuten, Inc.

REFERENCES

[1] C. Dellarocas, X. M. Zhang, and N. F. Awad, "Exploring the value of online product reviews in forecasting sales: The case of motion pictures," *Journal of Interactive Marketing*, vol. 2, no. 4, pp. 23-45, 2007.

[2] L. Zhu, G. Yin, and W. He, "Is this opinion leader's review useful? Peripheral cues for online review helpfulness," *Journal of Electronic Commerce Research*, vol. 15, no. 4, pp. 267-280, 2014.

[3] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, 1st ed. New York, USA.: Cambridge University Press, 2015, ch. 1, p. 3.

[4] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093-1113, 2014.

[5] H. Zhang, G. Wenyan, and J. Bo, “Machine learning and lexicon based methods for sentiment classification: A survey,” in *Proc. 11th Web Information System and Application Conference*, 2014, pp. 262-265.
[6] V. Gavrishchaka, Z. Yang, R. Miao, and O. Senyukova, “Advantages of hybrid deep learning frameworks in applications with limited data,” *International Journal of Machine Learning and Computing*, vol. 8, no. 6, pp. 549-558, 2018.
[7] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, pp. 1-25, 2018.
[8] X. Wang, W. Jiang, and Z. Luo, “Combination of convolutional and recurrent neural network for sentiment analysis of short texts,” in *Proc. COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2428-2437.
[9] M. Huang, Q. Qian, and X. Zhu, “Encoding syntactic knowledge in neural networks for sentiment classification,” *ACM Transactions on Information Systems (TOIS)*, vol. 35, no. 3, pp. 26:1-27, 2017.
[10] O. Appel, F. Chiclana, J. Carter, and H. Fujita, “A hybrid approach to the sentiment analysis problem at the sentence level,” *Knowledge-Based System*, vol. 108, pp. 110-124, 2016.
[11] S. Ebert, N. T. Vu, and H. Schütze, “A linguistically informed convolutional neural network,” in *Proc. the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2015, pp. 109-114.
[12] Z. Teng, D. T. Vo, and Y. Zhang, “Context-sensitive lexicon features for neural sentiment analysis,” in *Proc. the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1629-1638.
[13] C. Guggilla, T. Miller, and I. Gurevych, “CNN-and LSTM-based claim classification in online user comments,” in *Proc. the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2740-2751.
[14] H. Takamura, T. Inui, and M. Okumura, “Extracting semantic orientations of words using spin model,” in *Proc. the 43rd Annual Meeting on Association for Computational Linguistics*, 2005, pp. 133-140.
[15] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proc. the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532-1543.
[16] Z. Y. Cui, R. M. Ke, Z. Y. Pu, and Y. H. Wang, “Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction,’ *arXiv preprint arXiv:1801.02143*, 2018.

Mate Kovacs received his BA in Japanese studies from Karoli Gaspar University, Hungary. Since 2016, he is at the Graduate School of Information Science and Engineering, Ritsumeikan University, Japan. He received his master’s degree there in engineering, and he is currently pursuing his PhD in the same department. His research interests include statistical modeling of natural languages, deep learning, and data mining for business purposes.

Victor V. Kryssanov received his PhD from the Russian Academy of Sciences in 1994. He currently serves as a professor at the College of Information Science and Engineering, Ritsumeikan University, Japan. Before joining Ritsumeikan University in 2004, he was a JSTA researcher at Kyoto University, a JSPS research associate at Kobe University, and a NEDO guest researcher at the JSPMI Technical Research Institute, Tokyo. His research interests include social network analysis, communication theories, and statistical modeling of complex systems.