Department: Affective Computing and Sentiment Analysis
Editor: Erik Cambria, Nanyang Technological University

Sentiment and Sarcasm Classification With Multitask Learning

Navonil Majumder
Instituto Politécnico Nacional

Soujanya Poria
Nanyang Technological University

Haiyun Peng
Nanyang Technological University

Niyati Chhaya
Adobe Research

Erik Cambria
Nanyang Technological University

Alexander Gelbukh
Instituto Politécnico Nacional

***Abstract*—Sentiment classification and sarcasm detection are both important natural language processing tasks. Sentiment is always coupled with sarcasm where intensive emotion is expressed. Nevertheless, most literature considers them as two separate tasks. We argue that knowledge in sarcasm detection can also be beneficial to sentiment classification and vice versa. We show that these two tasks are correlated, and present a multitask learning-based framework using a deep neural network that models this correlation to improve the performance of both tasks in a multitask learning setting. Our method outperforms the state of the art by 3–4% in the benchmark dataset.**

The surge of Internet has enabled large-scale text-based opinion sharing on a wide range of topics. This has led to the opportunity of mining user sentiment on various subjects from the data publicly available over the Internet. The most important task in the analysis of users' opinions is sentiment classification: determining whether a given text, such as a user review, comment, or tweet, carries positive or negative polarity.

When expressing their opinions, users often use sarcasm for emphasizing their sentiment. In a sarcastic text, the sentiment intended by the author is the opposite of its literal meaning.

Digital Object Identifier 10.1109/MIS.2019.2904691
Date of current version 17 July 2019.

Published by the IEEE Computer Society

For example, the sentence "*Thank you alarm for never going off*" is literally positive ("*Thank you*"), however, the intended sentiment is negative "*alarm never going off*." Unless this sentiment shift is detected with semantics, the classifier may fail to spot sarcasm.

Currently, most researchers focus on either sentiment classification or sarcasm detection,[1,2] without considering the possibility of mutual influence between the two tasks. However, one can observe that the two tasks are correlated: people often use sarcasm as a device for the expression of emphatic negative sentiment. This observation can lead to a simple way in which one of the two tasks can help improve the other, i.e., if an expression can be detected as sarcastic, its sentiment can be assumed negative; if the expression can be classified as positive, then it can be assumed not sarcastic.

Here, we show that while this logic does lead to a slight improvement, there is a better way of combining the two tasks. Namely, in this paper, we train a classifier for both sarcasm and sentiment in a single neural network using multitask learning, a novel learning scheme that has gained recent popularity.[3,4] We empirically show that this method outperforms the results obtained with two separate classifiers and, in particular, outperforms the current state of the art by Mishra *et al.*[5]

The remainder of this paper is organized as follows: next section outlines related work; after that, we present our approach and list the baselines; next, results are discussed; finally, the last section concludes the paper.

RELATED WORK

Machine learning methods and deep neural networks, such as convolutional, recursive, recurrent, and memory networks, have shown good performance for sentiment detection.[6–9] Knowledge-based methods explore syntactic patterns[10] and employ sentiment resources.[11] However, sarcasm detection currently focuses on extracting features, such as syntactic,[12] surface pattern-based,[13] or personality-based features,[1] as well as contextual incongruity.[2]

Mishra *et al.*[5] extracted multimodal cognitive features for both sentiment classification and sarcasm detection, without modeling the two tasks in a single system. However, recently multitask learning has been successfully applied in many natural language processing tasks, such as implicit discourse relationship identification[4] and key-phrase boundary classification.[3] In this paper, we apply it to sentiment classification and sarcasm detection.

METHOD

According to Riloff *et al.*,[14] most sarcastic sentences carry negative sentiment. We leverage this to improve both sentiment classification and sarcasm detection. We use multitask learning, where a single neural network is used to perform more than one classification task (in our case, sentiment classification and sarcasm detection). This network facilitates synergy between the two tasks, resulting in improved performance on both tasks in comparison with their standalone counterparts (Fig. 1).

Task Definition

We solve two tasks with a single network. Given a sentence $[w_1, w_2, \ldots, w_l]$, where w_i are words, we assign it both a sentiment tag (positive/negative) and a sarcasm tag (yes/no).

Input Representation

We use D_g-dimensional ($D_g = 300$) Glove word-embeddings $x_i \in \mathbb{R}^{D_g}$ to represent the words w_i, padding the variable-length input sentences to a fixed length with null vectors. Thus, the input is represented as a matrix $X = [x_1, x_2, \ldots, x_L]$, where L is the length of the longest sentence.

Sentence Representation

In the next layers, we obtain sentence representation from X using gated recurrent unit (GRU) with attention mechanism as explained below.

Sentence-Level Word Representation The sentence X is fed to a GRU of size $D_{\text{gru}} = 500$ with parameters $W^{[z,r,h]} \in \mathbb{R}^{D_g \times D_{\text{gru}}}$ and $U^{[z,r,h]} \in \mathbb{R}^{D_{\text{gru}} \times D_{\text{gru}}}$ to get context-rich sentence-level word representations $H = [h_1, h_2, \ldots, h_L]$, $h_t \in \mathbb{R}^{D_{\text{gru}}}$ at the hidden output of the GRU.

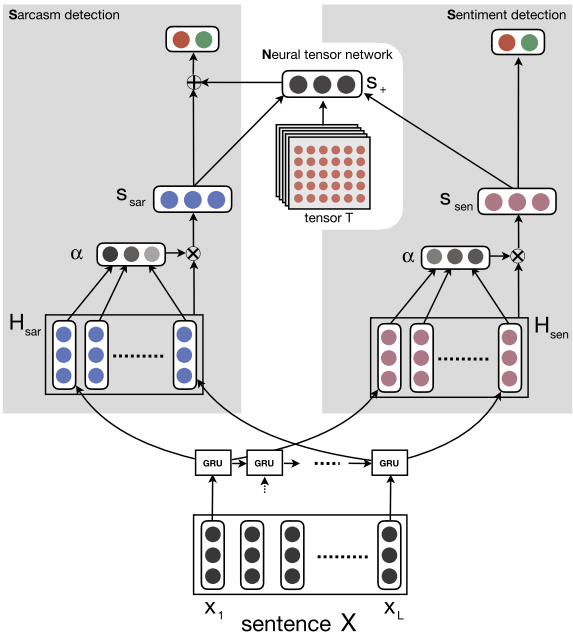

Figure 1. Our multitask architecture.

We use H for both sarcasm and sentiment. Thus, H is transformed to H_{sar} and H_{sen} using two different fully connected layers of size $D_t = 300$ in order to accommodate two different tasks, sarcasm detection, and sentiment classification

$$H_{\text{sar}} = ReLU(HW_{\text{sar}} + b_{\text{sar}})$$
$$H_{\text{sen}} = ReLU(HW_{\text{sen}} + b_{\text{sen}})$$

where $W_{[\text{sar,sen}]} \in \mathbb{R}^{D_{\text{dru}} \times D_t}$ and $b_{[\text{sar,sen}]} \in \mathbb{R}^{D_t}$.

Attention Network Word representations in H_* are encoded with task-specific sentence-level context. To aggregate these context-rich representations into the sentence representation s_*, we use attention mechanism, due to its ability to prioritize words relevant for the classification

$$P = \tanh(H_* W^{ATT}) \quad (1)$$

$$\alpha = \text{softmax}(P^T W^{\alpha}) \quad (2)$$

$$s_* = \alpha H_*^T \quad (3)$$

where $W^{ATT} \in \mathbb{R}^{D_t \times 1}$, $W^{\alpha} \in \mathbb{R}^{L \times L}$, $P \in \mathbb{R}^{L \times 1}$, and $s_* \in \mathbb{R}^{D_t}$. In (2), $\alpha \in [0,1]^L$ gives the relevance of words for the task, multiplied in (3) by the context-aware word representations in H_*.

Inter-Task Communication

We use neural tensor network (NTN) of size $D_{\text{ntn}} = 100$ to fuse sarcasm- and sentiment-specific sentence representations s_{sar} and s_{sen} to obtain the fused representation s_+, where

$$s_+ = \tanh(s_{\text{sar}} T^{[1:D_{\text{ntn}}]} s_{\text{sen}}^T + (s_{\text{sar}} \oplus s_{\text{sen}})W + b)$$

where $T \in \mathbb{R}^{D_{\text{ntn}} \times D_t \times D_t}$, $W \in \mathbb{R}^{2D_t \times D_{\text{ntn}}}$, $b, s_+ \in \mathbb{R}^{D_{\text{ntn}}}$, and $\oplus$ stands for concatenation. The vector s_+ contains information relevant to both sentiment and sarcasm. Instead of NTN, we also tried attention and concatenation for fusion, which resulted in inferior performance, as shown in the second last section.

Classification

For the two tasks, we use two different softmax layers for classifications.

Sentiment Classification We use only s_{sen} as sentence representation for sentiment classification, since we observe best performance without s_+. We apply softmax layer of size C ($C = 2$ for binary task) on s_{sen} for classification as follows:

$$\mathcal{P}_{\text{sen}} = \text{softmax}(s_{\text{sen}}\ W_{\text{sen}}^{\text{softmax}} + b_{\text{sen}}^{\text{softmax}})$$
$$\hat{y}_{\text{sen}} = \underset{j}{\text{argmax}}(\mathcal{P}_{\text{sen}}[j])$$

where $W_{\text{sen}}^{\text{softmax}} \in \mathbb{R}^{D_t \times C}$, $b_{\text{sen}}^{\text{softmax}} \in \mathbb{R}^C$, $\mathcal{P}_{\text{sen}} \in \mathbb{R}^C$, j is the class value (0 for negative and 1 for positive), and $\hat{y}_{\text{sen}}$ is the estimated class value.

Sarcasm Classification We use $s_{\text{sar}} \oplus s_+$ as sentence representation for sarcasm classification using softmax layer with size C ($C = 2$) as follows:

$$\mathcal{P}_{\text{sar}} = \text{softmax}((s_{\text{sar}} \oplus s_+)\ W_{\text{sar}}^{\text{softmax}} + b_{\text{sar}}^{\text{softmax}})$$
$$\hat{y}_{\text{sar}} = \underset{j}{\text{argmax}}(\mathcal{P}_{\text{sar}}[j])$$

where $W_{\text{sar}}^{\text{softmax}} \in \mathbb{R}^{(D_t + D_{ntn}) \times C}$, $b_{\text{sar}}^{\text{softmax}} \in \mathbb{R}^C$, $\mathcal{P}_{\text{sar}} \in \mathbb{R}^C$, j is the class value (0 for no and 1 for yes), and $\hat{y}_{\text{sar}}$ is the estimated class value.

Training

We use categorical cross entropy as the loss function (J_*; * is sar or sen) for training

$$J_* = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=0}^{C-1} y_{ij}^* \log \mathcal{P}_{*i}[j]$$

where N is the number of samples, i is the index of a sample, j is the class value, and

$$y_{ij}^{*} = \begin{cases} 1, \text{if expected class value of sample } i \text{is } j \\ 0, \text{otherwise.} \end{cases}$$

For training, we use ADAM,[15] an algorithm based on stochastic gradient descent which optimizes each parameter individually with different and adaptive learning rates. Also, we minimize both loss functions, namely J_{sen} and J_{sar}, with equal priority, by optimizing the parameter set

$$\theta = \{U^{[z,r,h]}, W^{[z,r,h]}, W_{*}, b_{*}, W^{ATT} \\ W^{\alpha}, T, W, b, W_{*}^{\text{softmax}}, b_{*}^{\text{softmax}}\}.$$

EXPERIMENTS

Dataset

The dataset[16] consists of 994 samples, each sample containing a text snippet labeled with sarcasm tag, sentiment tag, and eye-movement data of seven readers. We ignored the eye-movement data in our experiments. Of those samples, 383 are positive and 350 are sarcastic.

Baselines and Model Variants

We evaluated the following baselines and variations of our model.

Standalone Classifiers Here, we used

$$h_{*} = \text{FCLayer}(\text{GRU}(X))$$
$$\mathcal{P} = \text{SoftmaxLayer}(h_{*})$$

where * represents sar or sen and X is the input sentence as a list of word embeddings. We feed X to GRU and pass the final output through a fully connected layer (FCLayer) to obtain sentence representation h_{*}. We apply final softmax classification (SoftmaxLayer) to h_{*}.

Sentiment Coerced by Sarcasm In this classifier, the sentences classified as sarcastic are forced to be considered negative by the sentiment classifier.

Simple Multitask Classifier The following equations summarize this variant:

$$h_{*} = \text{FCLayer}_{*}(\text{GRU}(X)) \quad (4)$$

$$\mathcal{P}_{*} = \text{SoftmaxLayer}_{*}(h_{*}) \quad (5)$$

where * represents sar or sen. This setting shares the GRU between two tasks. Final output of the GRU is taken as the sentence representation. Sentence representation is fed to two different task-specific fully connected layers (FCLayer_{*}), giving h_{*}. Subsequently, h_{*} are fed to two different softmax layers SoftmaxLayer_{*} for classification.

Simple Multitask Classifier With Fusion In this variant, we changed (5) to

$$\mathcal{P}_{\text{sar}} = \text{SoftmaxLayer}_{\text{sar}}(h_{\text{sar}} \oplus F) \quad (6)$$

$$\mathcal{P}_{\text{sen}} = \text{SoftmaxLayer}_{\text{sen}}(h_{\text{sen}}) \quad (7)$$

where $F = \text{NTN}(h_{\text{sar}}, h_{\text{sen}})$. Here, h_{sar} and h_{sem} are fed to a NTN whose output is concatenated with h_{sar} for classification. Sentiment classification is done with h_{sen} only. We also tried variants with other methods of fusion (such as fully connected layer or Hadamard product) instead of NTN, as well as variants with $h_{\text{sen}} \oplus F$ instead of, or in addition to, $h_{\text{sar}} \oplus F$, but they did not improve the results.

Task-Specific GRU With Fusion Here, we used two separate GRUs for the two tasks in (4)

$$h_{*} = \text{FCLayer}_{*}(\text{GRU}_{*}(X)). \quad (8)$$

We used (6) and (7) for $\mathcal{P}_{*}$. Again, we tried concatenating F with h_{sen}, both, or none as in (5), but this did not improve the results.

Best Model: Shared Attention Here, we added the attention mechanism to the matrix H_{*} in (4), and used (6) and (7) for $\mathcal{P}_{*}$. This model, described in detail in the previous section, is the main model we present in this paper since it gave the best results. We also tried separate GRUs as in (8), but this did not improve the results.

RESULTS AND DISCUSSION

The results using tenfold cross validation are shown in Table 1. As baselines, we used the standalone sentiment and sarcasm classifiers, as well as the CNN-based state-of-the-art method by Mishra *et al.*[5] Our standalone GRU-based sentiment and sarcasm classifiers performed slightly better than the state of the art, even though this also uses the gaze data present in the dataset but this is hardly available in any real-life setting.

Table 1. Results for various experiments.

Variant	Sentiment			Sarcasm			Average
	Precision	Recall	F-Score	Precision	Recall	F-Score	F-Score
State of the art[5]	79.89	74.86	77.30	87.42	87.03	86.97	82.13
Standalone classifiers	79.02	78.03	78.13	89.96	89.25	89.37	83.75
Standalone coerced	81.57	80.06	80.38	–	–	–	–
Multi-Task simple	80.41	79.88	79.7	89.42	89.19	89.04	84.37
Multi-Task with fusion	82.32	81.71	81.53	90.94	**90.74**	**90.67**	86.10
Multi-Task with fusion and separate GRUs	80.54	80.02	79.86	**91.01**	90.66	90.62	85.24
Multi-Task with fusion and shared attention (Section 2)	**83.67**	**83.10**	**83.03**	90.50	90.34	90.29	**86.66**

In contrast, our method, besides improving results, is applied to plain-text documents such as tweets, without any gaze data.

As expected, the sentiment classifier coerced by sarcasm classifier performed better than the standalone sentiment classifier. This means that an efficient sarcasm detector can boost the performance of a sentiment classifier. All our multitask classifiers outperformed both standalone classifiers. However, the margin of improvement for multitask classifier over the standalone classifier is greater for sentiment than for sarcasm. Probably this is because sarcasm detection is a subtask of sentiment analysis.[17]

Analyzing examples and attention visualization of the multitask network, we observed that the multitask network mainly helps improving sarcasm classification when there is a strong sentiment shift, which indicates the possibility of sarcasm in the sentence. The example given in the introduction was classified incorrectly by the standalone sarcasm classifier but correctly by the standalone sentiment classifier; coercing one of the classifiers by the other would not change the result. In the multitask network, both sentiment and sarcasm are detected correctly, apparently because the network detected the sentiment shift in the sentence, which improved sarcasm classification.

Similarly, the sentence "*Absolutely love when water is spilt on my phone, just love it*" is classified as positive by the standalone sentiment classifier: "*Absolutely love*" highlighted by the attention scores (not presented in this short paper). However, the standalone sarcasm classifier identified it as sarcastic due to "*water spilt on my phone*" (seen from the attention scores) and in the multitask network this clue corrected the sentiment classifier's output.

Even our standalone GRU-based classifiers outperformed the CNN-based state-of-the-art method. The multitask classifiers outperformed the standalone classifiers because of the shared representation, which serves as additional regularization for each task from the other task.

Adding NTN fusion to the multitask classifier further improved results, giving the best performance for sarcasm detection. Adding an attention network shared between the tasks further improves the performance for sentiment classification. As the last column of Table 1 shows, on average, the best results across the two tasks were obtained with the architecture described in the second section.

CONCLUSION

We presented a classifier architecture that can be trained on sentiment or sarcasm data and outperforms the state of the art in both cases on the dataset used by Mishra *et al.*[5] Our architecture uses a GRU-based neural network, while the state-of-the-art method used a CNN.

Furthermore, we showed that multitask learning-based methods significantly outperform standalone sentiment and sarcasm classifiers. This indicates that sentiment classification and sarcasm detection are related tasks.

Finally, we presented a multitask learning architecture that gave the best results, out of a number of variants of the architecture that we tried.

To make our claim more robust, we plan to build a new dataset for rigorous experimentation. In addition, we intend to incorporate multimodal information in our network for enhancing its performance.

REFERENCES

1. S. Poria *et al.*, "A deeper look into sarcastic tweets using deep convolutional neural networks," in *Proc. 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 1601–1612.

2. A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing context incongruity for sarcasm detection," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. (Volume 2: Short Papers)*, 2015, pp. 757–762.
3. I. Augenstein and A. Sŀgaard, "Multi-task learning of keyphrase boundary classification," in *Proc. 55th Annual Meeting Assoc. Comput. Linguistics (Volume 2: Short Papers)*, 2017, pp. 341–346.
4. M. Lan *et al.*, "Multi-task attention-based neural networks for implicit discourse relationship representation and identification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1299–1308.
5. A. Mishra, K. Dey, and P. Bhattacharyya, "Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, 2017, pp. 377–387.
6. A. Zadeh *et al.*, "Multi-attention recurrent network for human communication comprehension," in *Proc. Assoc. Advancement Artif. Intell.*, 2018, pp. 5642–5649.
7. N. Majumder *et al.*, "DialogueRNN: An attentive RNN for emotion detection in conversations," in *Proc. Assoc. Advancement Artif. Intell.*, 2019.
8. L. Dong *et al.*, "Adaptive recursive neural network for target-dependent twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Volume 2: Short Papers)*, 2014, pp. 49–54.
9. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
10. S. Poria *et al.*, "Sentiment data flow analysis by means of dynamic linguistic patterns," *IEEE Comput. Intell. Mag.*, vol. 10, no. 4, pp. 26–36, Nov. 2015.
11. E. Cambria *et al.*, "SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1795–1802.
12. F. Barbieri, H. Saggion, and F. Ronzano, "Modelling sarcasm in twitter, a novel approach," in *Proc. 5th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2014, pp. 50–58.
13. D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in Twitter and Amazon," in *Proc. 14th Conf. Comput. Natural Lang. Learn.*, 2010, pp. 107–116.
14. E. Riloff *et al.*, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 704–714.
15. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. 3rd Int. Conf. Learn. Representation*, 2014.
16. A. Mishra, D. Kanojia, and P. Bhattacharyya, "Predicting readers' sarcasm understandability by modeling gaze behavior," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3747–3753.
17. E. Cambria *et al.*, "Sentiment analysis is a big suitcase," *IEEE Intell. Syst.*, vol. 32, no. 6, pp. 74–80, Nov.–Dec. 2017.

Navonil Majumder is currently working toward the Ph.D. degree at Instituto Politécnico Nacional, Mexico. His research interests include natural language processing, machine learning, neural networks and deep learning. Contact him at navo@nlp.cic.ipn.mx.

Soujanya Poria is a presidential research fellow at Nanyang Technological University, Singapore. His research interests include in sentiment analysis, multimodal interaction, natural language processing, and affective computing. He received the Ph.D. degree in Computer Science and Mathematics from the University of Stirling. Contact him at sporia@ntu.edu.sg.

Haiyun Peng is currently working toward the Ph.D. degree at Nanyang Technological University, Singapore. His research interests include multilingual sentiment analysis, text representation learning, dialogue system, and deep learning. Contact him at peng0065@ntu.edu.sg.

Niyati Chhaya is a senior research scientist at Adobe Research, India. Her research interests include natural language processing and machine learning with a current research focus on affective content analysis. She received the Ph.D. degree from the University of Maryland. Contact her at nchhaya@adobe.com.

Erik Cambria is an associate professor at Nanyang Technological University, Singapore. His research interests include natural language understanding, commonsense reasoning, sentiment analysis, and multimodal interaction. He is the corresponding author and can be contacted at cambria@ntu.edu.sg.

Alexander Gelbukh is currently a professor at Instituto Politécnico Nacional, Mexico. His research interests include artificial intelligence, computational linguistics, sentiment and emotion analysis, computational morphology, etc. He received the Ph.D. degree in computer science from All-Russian Institute for Scientific and Technical Information, Moscow, Russia. Contact him at gelbukh@cic.ipn.mx.