Sentiment and Sarcasm Classification with Multitask Learning

Sentiment classification and sarcasm detection are both important natural language processing (NLP) 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 multi-task learning-based framework using a deep neural network that models this correlation to improve the performance of both tasks in a multi-task 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. 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, 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 multi-task learning, a novel learning scheme that has gained recent popularity. 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.

The remainder of the paper is organized as follows: Section 1 outlines related work; Section 2 presents our approach; Section 3 lists the baselines; Section 4 discusses results; finally, Section 5 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. Knowledge-based methods explore syntactic patterns and employ sentiment resources. However, sarcasm detection currently focuses on extracting features, such as syntactic, surface pattern-based, or personality-based features, as well as contextual incongruity.

Mishra et al. extracted multimodal cognitive features for both sentiment classification and sarcasm detection, without modeling the two tasks in a single system. However, recently multi-task learning has been successfully applied in many NLP tasks, such as implicit discourse relationship identification and key-phrase boundary classification. In this paper, we apply it to sentiment classification and sarcasm detection.

### METHOD

According to Riloff et al., most sarcastic sentences carry negative sentiment. We leverage this to improve both sentiment classification and sarcasm detection. We use multi-task 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.

**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 \( D_y \)-dimensional (\( D_y = 300 \)) Glove word-embeddings \( x_i \in \mathbb{R}^{D_y} \) 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.
Figure 1. Our multi-task architecture.

**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 $D_{gru} = 500$ with parameters $W^{[z, r, h]} \in \mathbb{R}^{D_{gru} \times D_{gru}}$ and $U^{[z, r, h]} \in \mathbb{R}^{D_{gru} \times D_{gru}}$ to get context-rich sentence-level word representations $H = [h_1, h_2, \cdots, h_L], h_t \in \mathbb{R}^{D_{gru}}$ at the hidden output of the GRU.

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:

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

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

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

$$P = \tanh(H_s W^{ATT}),$$
$$\alpha = \text{softmax}(PT W^\alpha),$$
$$s_s = \alpha H_s^T,$$

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

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

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

where $T \in \mathbb{R}^{D_{ntn} \times D_s \times D_t}$, $W \in \mathbb{R}^{D_t \times D_{ntn}}$, $b, s_e \in \mathbb{R}^{D_{ntn}}$, and $\oplus$ stands for concatenation. The vector $s_e$ contains information relevant to both sentiment and sarcasm. Instead of NTN, we also tried attention and concatenation for fusion, which resulted in inferior performance (Section 4).

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 $s_e$. We apply softmax layer of size $C$ ($C = 2$ for binary task) on $s_{sen}$ for classification as follows:

$$P_{sen} = \text{softmax}(s_{sen} W_{softmax}^{sen} + b_{softmax}^{sen}),$$

$$\hat{y}_{sen} = \arg\max_j (P_{sen}[j]),$$

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

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

$$P_{sar} = \text{softmax}((s_{sar} \oplus s_e) W_{softmax}^{sar} + b_{softmax}^{sar}),$$

$$\hat{y}_{sar} = \arg\max_j (P_{sar}[j]),$$

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

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

$$J_* = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=0}^{C-1} y^*_{ij} \log P_*[j],$$

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\textsuperscript{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

$$\theta = \{U^{[z,r,h]}, W^{[z,r,h]}, W', b', W^{ATT}, W^{\alpha}, T, W, b, W_{softmax}^{sar}, b_{softmax}^{sar}\}.$$
EXPERIMENTS

Dataset  The dataset consists of 994 samples, each sample containing a text snippet labeled with sarcasm tag, sentiment tag, and eye-movement data of 7 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

\[ h^* = \text{FCLayer}(\text{GRU}(X)) , \]
\[ P^* = \text{SoftmaxLayer}(h^*) , \]

where * represents sar or sen, \( X \) is the input sentence as a list of word embeddings. We feed \( X \) to \( \text{GRU} \) and pass the final output through a fully-connected layer (\( \text{FCLayer} \)) to obtain sentence representation \( h^* \). We apply final softmax classification (\( \text{SoftmaxLayer} \)) 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 multi-task classifier  The following equations summarize this variant:

\[ h^* = \text{FCLayer}^*(\text{GRU}(X)) , \]
\[ P^* = \text{SoftmaxLayer}^*(h^*) , \]

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

Simple multi-task classifier with fusion  In this variant, we changed Eq. (5) to:

\[ \mathcal{P}_{\text{sar}} = \text{SoftmaxLayer}_{\text{sar}}(h_{\text{sar}} \oplus F) , \]
\[ \mathcal{P}_{\text{sen}} = \text{SoftmaxLayer}_{\text{sen}}(h_{\text{sen}}) , \]

where \( F = \text{NTN}(h_{\text{sar}}, h_{\text{sen}}) \). Here, \( h_{\text{sar}} \) and \( h_{\text{sen}} \) are fed to a NTN whose output is concatenated with \( h_{\text{sar}} \) for classification. Sentiment classification is done with \( h_{\text{sen}} \) 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 Eq. (4):

\[ h^* = \text{FCLayer}^*(\text{GRU}^*(X)) . \]

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

Table 1. Results for various experiments.

| Variant Sentiment | Sentiment | Sarcasm | Sarcasm | Average | Average |
|-------------------|-----------|---------|---------|---------|---------|
|                   | Precision | Recall  | F-Score | Precision | Recall  | F-Score | Precision | Recall  | F-Score |
| State of the art  | 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 | 83.75 |
| Multi-Task with fusion | 82.32 | 81.71 | 81.53 | 90.04 | 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 |

Best model: shared attention

Here, we added the attention mechanism to the matrix $H_*$ in Eq. (4), and used Eq. (6) and Eq. (7) for $P_*$. This model, described in detail in Section 2, is the main model we present in this paper since it gave the best results. We also tried separate GRUs as in Eq. (5), but this did not improve the results.

RESULTS AND DISCUSSION

The results using 10-fold 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. 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. 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 multi-task classifiers outperformed both standalone classifiers. However, the margin of improvement for multi-task classifier over the standalone classifier is greater for sentiment than for sarcasm. Probably this is because sarcasm detection is a subtask of sentiment analysis.

Analyzing examples and attention visualization of the multi-task network, we observed that the multi-task 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 multi-task 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 multi-task 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 multi-task 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 multi-task 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 Section 2.
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. Our architecture uses a GRU-based neural network, while the state-of-the-art method used a CNN.

Furthermore, we showed that multi-task 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 multi-task 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,” COLING, 2016, pp. 1601–1612.
2. A. Joshi, V. Sharma, and P. Bhattacharyya, “Harnessing Context Incongruity for Sarcasm Detection,” ACL, 2015, pp. 757–762.
3. I. Augenstein and A. Søgaard, “Multi-Task Learning of Keyphrase Boundary Classification,” ACL, 2017, pp. 341–346.
4. M. Lan et al., “Multi-task Attention-based Neural Networks for Implicit Discourse Relationship Representation and Identification,” EMNLP, 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,” ACL, 2017, pp. 377–387.
6. A. Zadeh et al., “Multi-attention recurrent network for human communication comprehension,” AAAI, 2018, pp. 5642–5649.
7. N. Majumder et al., “DialogueRNN: An Attentive RNN for Emotion Detection in Conversations,” AAAI, 2019.
8. L. Dong et al., “Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification,” ACL, 2014, pp. 49–54.
9. I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” NIPS, 2014, pp. 3104–3112.
10. S. Poria et al., “Sentiment data flow analysis by means of dynamic linguistic patterns,” IEEE Computational Intelligence Magazine, vol. 10, no. 4, 2015, pp. 26–36.
11. E. Cambria et al., “SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings,” AAAI, 2018, pp. 1795–1802.
12. F. Barbieri, H. Saggion, and F. Ronzano, “Modelling Sarcasm in Twitter, a Novel Approach,” WASSA@ACL, 2014, pp. 50–58.
13. D. Davidov, O. Tsur, and A. Rappoport, “Semi-supervised recognition of sarcastic sentences in Twitter and Amazon,” CoNLL, 2010, pp. 107–116.
14. E. Riloff et al., “Sarcasm as Contrast between a Positive Sentiment and Negative Situation,” EMNLP, 2013, pp. 704–714.
15. D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” CoRR, vol. abs/1412.6980, 2014.
16. A. Mishra, D. Kanojia, and P. Bhattacharyya, “Predicting Readers’ Sarcasm Understandability by Modeling Gaze Behavior,” AAAI, 2016, pp. 3747–3753.
17. E. Cambria et al., “Sentiment Analysis is a Big Suitcase,” IEEE Intelligent Systems, vol. 32, no. 6, 2017, pp. 74–80.

ABOUT THE AUTHORS

Navonil Majumder is a PhD Student 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 lie in sentiment analysis, multimodal interaction, natural language processing, and affective computing. Poria received a PhD in Computer Science and Mathematics from University of Stirling. Contact him at sporia@ntu.edu.sg.

Haiyun Peng is a PhD Student 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 are in natural language processing and machine learning with a current research focus on affective content analysis. Chhaya completed her PhD 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. Contact him at cambria@ntu.edu.sg.

Alexander Gelbukh is a professor at Instituto Politécnico Nacional, Mexico. His research interests lie in artificial intelligence, computational linguistics, sentiment and emotion analysis, computational morphology etc. Gelbukh received a PhD in Computer Science from All-Russian Institute for Scientific and Technical Information (VINITI). Contact him at gelbukh@cic.ipn.mx.