A Multi-Task Learning Approach to Sarcasm Detection (Student Abstract)

Edoardo Savini, Cornelia Caragea
Department of Computer Science, University of Illinois at Chicago
{esavin3, cornelia}@uic.edu

Abstract

Sarcasm detection plays an important role in natural language processing as it has been considered one of the most challenging subtasks in sentiment analysis and opinion mining applications. Our work aims to detect sarcasm in social media sites and discussion forums, exploiting the potential of deep neural networks and multi-task learning. Specifically, relying on the strong correlation between sarcasm and (implied negative) sentiment, we explore a multi-task learning framework that uses sentiment classification as an auxiliary task to inform the main task of sarcasm detection. Our proposed model outperforms many previous baseline methods on an existing large dataset annotated with sarcasm.

Introduction

Sarcasm can be defined as a sophisticated linguistic phenomenon (and a form of speech act) that makes use of figurative images to implicitly convey contempt through incongruity between text and context (Joshi, Sharma, and Bhattacharyya 2015). Its highly figurative nature has caused it to be considered as one of the main challenges in sentiment analysis and opinion mining applications. While many previous works on this task have focused on approaches based on feature engineering, that use distant (or weak) supervision and Support Vector Machines to extract lexical cues recurrent in sarcasm, we continue the path followed by (Ghosh and Veale 2016; Joshi et al. 2016) in attempting to automatically detect sarcasm harnessing the potentials of deep neural networks combined with word embeddings to capture both semantic and syntactic features in sarcastic utterances.

Given that sarcastic statements are usually associated with (implied) negative sentiment, we attempt to exploit this correlation using a multi-task learning framework to improve the performance of sarcasm detection by combining it with sentiment classification. A similar idea was developed recently by Majumder et al., who used a Gated Recurrent Unit model with attention mechanism and applied it on the dataset of about a thousand samples of sentences with both sarcastic and sentiment tags. In our work, we develop an architecture that exploits the combination of neural networks (in particular, Bidirectional Long Short Term Memory networks, BiLSTM, with an embedding layer) with sentiment classification, and evaluate its performance on a dataset of millions of self-annotated sentences from Reddit (SARC) introduced by Khodak, Saunshi, and Vodrahalli (2017). The state-of-the-art of this dataset was reached by Hazarika et al. (2018), who proposed a framework able to detect contextual information with user embedding created through user profiling and discourse modeling from comments on Reddit. In contrast, our model is able to predict sarcasm from the sentence itself, without considering contextual information such as user embedding. We show that our model outperforms many previous baselines that do not use user profiling and reaches a performance similar to the state-of-the-art that integrates user embeddings.

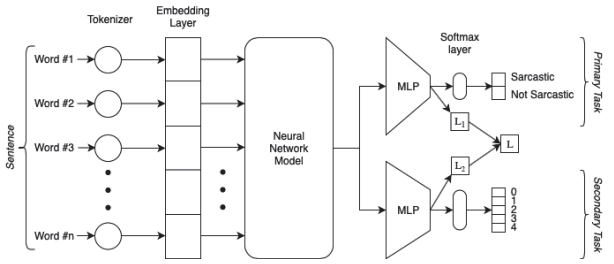

Figure 1: Our multi-tasking framework.

Proposed Model

The purpose of our model is to enhance the performance of the sarcasm detection task by adding an auxiliary task on sentiment classification. Figure 1 shows the multitasking framework we used for our experiment. We can see that both tasks share the same model (BiLSTM) and embedding, but each of them has its own Multi Layer Perceptron.

The configuration of our framework allows the secondary task to inform the training on the primary task when computing the loss of the model. Denoting Ω_1 as the sarcastic labeled dataset and Ω_2 as the sentiment labeled dataset, the total loss L of our framework is computed with the formula:

$$L = \sum_{(x,y)\in\Omega_1} L_1(x,y) + \lambda \sum_{(x,y)\in\Omega_2} L_2(x,y), \qquad (1)$$

where L_i is the cross-entropy loss computed for each sentence in the dataset Ω_i, between the desired output y and

Set	Sarcastic	Non-Sarcastic	Total
Original Train	505,390	505,390	1,010,780
New Train	404,312	404,312	808,624
Validation	101,078	101,078	202,156
Test	125,804	125,804	251,608

Table 1: SARC Main Balanced dataset size.

the output predicted by our network. The index $i \in \{1, 2\}$ distinguishes the primary task ($i = 1$) from the auxiliary ($i = 2$) one. The term λ is a hyper-parameter that limits the impact of the secondary task on the training parameters. In every epoch, our algorithm computes the gradient of L for each batch and propagates it to tune our model parameters through the AdaGrad optimizer.

In order to add a sentiment label to our dataset, we employed the pre-trained model for sentiment analysis available on Stanford NLP website.[1] The algorithm scans every text content one sentence at a time, assigning to each one an integer score from 0 to 4, that corresponds to the labels: Very Negative (0), Negative (1), Neutral (2), Positive (3), and Very Positive (4).

Experiments and Results

Dataset and Evaluation Setting: For our experiments, we used the main balanced variant of SARC dataset and the train/test split made available by Khodak, Saunshi, and Vodrahalli (2017). This dataset contains more than a million sarcastic and non-sarcastic self-annotated statements retrieved from Reddit. We divided the original training set into 80% training and 20% validation, and kept the test set as provided by the authors. Both the new train and validation sets have been shuffled and maintained balanced. Table 1 shows the size of each of the subsets used in our experiments.

Results and Observations: Table 2 shows the performance of BiLSTM with and without multi-task learning (MTL) (first block), as well as the comparison with several previous works that use either only the text or the combination of text and user information (second and third blocks, respectively). Both BiLSTM+MTL and BiLSTM are trained with the concatenation of contextual (ELMo) and non-contextual (FastText) word embeddings, which showed the best results over individual embeddings and embedding types. As we can see from Table 2, the BiLSTM that uses multi-tasking (BiLSTM+MTL) obtains slightly better performance as compared with BiLSTM without multi-tasking.

Next, we contrast BiLSTM+MTL trained with the concatenation of contextual (ELMo) and non-contextual (FastText) word embeddings with respect to other SARC's baseline models. Specifically, we compared our best model with the baselines examined by Hazarika et al. (2018) on the main balanced version of the SARC dataset. It can be observed that our model outperforms all the other models that do not use personality features (Bag-of-words, CNN, CASCADE) by about 10%. Interestingly, BiLSTM+MTL with no user information outperforms even CNN-SVM and CUE-CNN that model user embeddings by about 7%. Our F1-score is only 0.7% lower than the current state-of-the-art CASCADE that makes use of user personality features. We expect that augmenting our framework with user embeddings, to take into account personality features or other contextual information, could outperform the CASCADE model that uses user embeddings.

[1] https://nlp.stanford.edu/sentiment/

Models	F1-score
BiLSTM + MTL (proposed model)	**0.763**
BiLSTM	0.748
Bag-of-words (Hazarika et al. 2018)	0.64
CNN (Hazarika et al. 2018)	0.66
CASCADE (Hazarika et al. 2018) (no personality features)	0.66
CNN-SVM (Poria et al. 2016)	0.68
CUE-CNN (Amir et al. 2016)	0.69
CASCADE (Hazarika et al. 2018) (with personality features)	**0.77**

Table 2: The performance of BiLSTM+MTL and its comparison with previous works on SARC.

Conclusion and Future Work

We proposed a multitask learning framework to exploit the relation between sarcasm and the (implied) negative sentiment that a message conveys and demonstrated that the addition of the sentiment detection improves the effectiveness of the models for sarcasm detection. Our BiLSTM with Multitasking outperforms many previous baselines that do not exploit user embeddings on an existing dataset for sarcasm. However, based on an error analysis performed on the test set, we strongly believe that further improvements to the model could be obtained in the future by explicitly capturing (anti-correlated) sentiment and sarcasm, which could stand as an interesting future direction.

References

Amir, S.; Wallace, B. C.; Lyu, H.; and Silva, P. C. M. J. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976*.

Ghosh, A., and Veale, T. 2016. Fracking sarcasm using neural network. In *Workshop on computational approaches to subjectivity, sentiment and social media analysis*.

Hazarika, D.; Poria, S.; Gorantla, S.; Cambria, E.; Zimmermann, R.; and Mihalcea, R. 2018. Cascade: Contextual sarcasm detection in online discussion forums. In *COLING*.

Joshi, A.; Tripathi, V.; Patel, K.; Bhattacharyya, P.; and Carman, M. 2016. Are word embedding-based features useful for sarcasm detection? *arXiv preprint arXiv:1610.00883*.

Joshi, A.; Sharma, V.; and Bhattacharyya, P. 2015. Harnessing context incongruity for sarcasm detection. In *ACL*.

Khodak, M.; Saunshi, N.; and Vodrahalli, K. 2017. A large self-annotated corpus for sarcasm. *CoRR* abs/1704.05579.

Majumder, N.; Poria, S.; Peng, H.; Chhaya, N.; Cambria, E.; and Gelbukh, A. F. 2019. Sentiment and sarcasm classification with multitask learning. *CoRR* abs/1901.08014.

Poria, S.; Cambria, E.; Hazarika, D.; and Vij, P. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*.