Reasoning with Sarcasm by Reading In-between

Yi Tay[†], **Luu Anh Tuan**[ψ], **Siu Cheung Hui**[φ], **Jian Su**[δ]
[†]ytay017@e.ntu.edu.sg
[ψ]at.luu@i2r.a-star.edu.sg
[φ]asschui@ntu.edu.sg
[δ]sujian@i2r.a-star.edu.sg
[†,φ]School of Computer Science and Engineering, Nanyang Technological University
[ψ,δ]A*Star, Institute for Infocomm Research, Singapore

Abstract

Sarcasm is a sophisticated speech act which commonly manifests on social communities such as Twitter and Reddit. The prevalence of sarcasm on the social web is highly disruptive to opinion mining systems due to not only its tendency of polarity flipping but also usage of figurative language. Sarcasm commonly manifests with a contrastive theme either between positive-negative sentiments or between literal-figurative scenarios. In this paper, we revisit the notion of modeling *contrast* in order to reason with sarcasm. More specifically, we propose an attention-based neural model that looks *in-between* instead of *across*, enabling it to explicitly model contrast and incongruity. We conduct extensive experiments on six benchmark datasets from Twitter, Reddit and the Internet Argument Corpus. Our proposed model not only achieves state-of-the-art performance on all datasets but also enjoys improved interpretability.

1 Introduction

Sarcasm, commonly defined as *'An ironical taunt used to express contempt'*, is a challenging NLP problem due to its highly figurative nature. The usage of sarcasm on the social web is prevalent and can be frequently observed in reviews, microblogs (*tweets*) and online forums. As such, the battle against sarcasm is also regularly cited as one of the key challenges in sentiment analysis and opinion mining applications (Pang et al., 2008). Hence, it is both imperative and intuitive that effective sarcasm detectors can bring about numerous benefits to opinion mining applications.

Sarcasm is often associated to several linguistic phenomena such as (1) an explicit contrast between sentiments or (2) disparity between the conveyed emotion and the author's situation (context). Prior work has considered sarcasm to be a contrast between a positive and negative sentiment (Riloff et al., 2013). Consider the following examples:

1. I absolutely *love* to be *ignored*!
2. Yay!!! The best thing to wake up to is my neighbor's drilling.
3. Perfect movie for people who can't fall asleep.

Given the examples, we make a crucial observation - Sarcasm relies a lot on the semantic relationships (and contrast) between individual words and phrases in a sentence. For instance, the relationships between phrases {*love*, *ignored*}, {*best*, *drilling*} and {*movie*, *asleep*} (in the examples above) richly characterize the nature of sarcasm conveyed, i.e., word pairs tend to be contradictory and more often than not, express a juxtaposition of positive and negative terms. This concept is also explored in (Joshi et al., 2015) in which the authors refer to this phenomena as *'incongruity'*. Hence, it would be useful to capture the relationships between selected word pairs in a sentence, i.e., *looking in-between*.

State-of-the-art sarcasm detection systems mainly rely on deep and *sequential* neural networks (Ghosh and Veale, 2016; Zhang et al., 2016). In these works, compositional encoders such as gated recurrent units (GRU) (Cho et al., 2014) or long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) are often employed, with the input document being parsed one word at a time. This has several shortcomings for the sarcasm detection task. Firstly, there is

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 1010–1020
Melbourne, Australia, July 15 - 20, 2018.

no explicit interaction between word pairs, which hampers its ability to explicitly model contrast, incongruity or juxtaposition of situations. Secondly, it is difficult to capture long-range dependencies. In this case, contrastive situations (or sentiments) which are commonplace in sarcastic language may be hard to detect with simple sequential models.

To overcome the weaknesses of standard sequential models such as recurrent neural networks, our work is based on the intuition that modeling intra-sentence relationships can not only improve classification performance but also pave the way for more explainable neural sarcasm detection methods. In other words, our key intuition manifests itself in the form of an attention-based neural network. While the key idea of most neural attention mechanisms is to focus on relevant words and sub-phrases, it merely looks *across* and does not explicitly capture word-word relationships. Hence, it suffers from the same shortcomings as sequential models.

In this paper, our aim is to combine the effectiveness of state-of-the-art recurrent models while harnessing the intuition of *looking in-between*. We propose a multi-dimensional intra-attention recurrent network that models intricate similarities between each word pair in the sentence. In other words, our novel deep learning model aims to capture *'contrast'* (Riloff et al., 2013) and *'incongruity'* (Joshi et al., 2015) within end-to-end neural networks. Our model can be thought of self-targeted co-attention (Xiong et al., 2016), which allows our model to not only capture word-word relationships but also long-range dependencies. Finally, we show that our model produces interpretable attention maps which aid in the explainability of model outputs. To the best of our knowledge, our model is the first attention model that can produce explainable results in the sarcasm detection task.

Briefly, the prime contributions of this work can be summarized as follows:

- We propose a new state-of-the-art method for sarcasm detection. Our proposed model, the Multi-dimensional Intra-Attention Recurrent Network (MIARN) is strongly based on the intuition of compositional learning by leveraging intra-sentence relationships. To the best of our knowledge, none of the existing state-of-the-art models considered exploiting intra-sentence relationships, solely relying on sequential composition.
- We conduct extensive experiments on multiple benchmarks from Twitter, Reddit and the Internet Argument Corpus. Our proposed MIARN achieves highly competitive performance on all benchmarks, outperforming existing state-of-the-art models such as GRNN (Zhang et al., 2016) and CNN-LSTM-DNN (Ghosh and Veale, 2016).

2 Related Work

Sarcasm is a complex linguistic phenomena that have long fascinated both linguists and NLP researchers. After all, a better computational understanding of this complicated speech act could potentially bring about numerous benefits to existing opinion mining applications. Across the rich history of research on sarcasm, several theories such as the Situational Disparity Theory (Wilson, 2006) and the Negation Theory (Giora, 1995) have emerged. In these theories, a common theme is a motif that is strongly grounded in contrast, whether in sentiment, intention, situation or context. (Riloff et al., 2013) propagates this premise forward, presenting an algorithm strongly based on the intuition that sarcasm arises from a juxtaposition of positive and negative situations.

2.1 Sarcasm Detection

Naturally, many works in this area have treated the sarcasm detection task as a standard text classification problem. An extremely comprehensive overview can be found at (Joshi et al., 2017). Feature engineering approaches were highly popular, exploiting a wide diverse range of features such as syntactic patterns (Tsur et al., 2010), sentiment lexicons (González-Ibánez et al., 2011), n-gram (Reyes et al., 2013), word frequency (Barbieri et al., 2014), word shape and pointedness features (Ptáček et al., 2014), readability and flips (Rajadesingan et al., 2015), etc. Notably, there have been quite a reasonable number of works that propose features based on similarity and contrast. (Hernández-Farías et al., 2015) measured the Wordnet based semantic similarity between words. (Joshi et al., 2015) proposed a framework based on explicit and implicit incongruity, utilizing features based on positive-negative patterns. (Joshi et al., 2016) proposed similarity features based on word embeddings.

2.2 Deep Learning for Sarcasm Detection

Deep learning based methods have recently garnered considerable interest in many areas of NLP research. In our problem domain, (Zhang et al., 2016) proposed a recurrent-based model with a gated pooling mechanism for sarcasm detection on Twitter. (Ghosh and Veale, 2016) proposed a convolutional long-short-term memory network (CNN-LSTM-DNN) that achieves state-of-the-art performance.

While our work focuses on document-only sarcasm detection, several notable works have proposed models that exploit personality information (Ghosh and Veale, 2017) and user context (Amir et al., 2016). Novel methods for sarcasm detection such as gaze / cognitive features (Mishra et al., 2016, 2017) have also been explored. (Peled and Reichart, 2017) proposed a novel framework based on neural machine translation to convert a sequence from sarcastic to non-sarcastic. (Felbo et al., 2017) proposed a layer-wise training scheme that utilizes emoji-based distant supervision for sentiment analysis and sarcasm detection tasks.

2.3 Attention Models for NLP

In the context of NLP, the key idea of neural attention is to soft select a sequence of words based on their relative importance to the task at hand. Early innovations in attentional paradigms mainly involve neural machine translation (Luong et al., 2015; Bahdanau et al., 2014) for aligning sequence pairs. Attention is also commonplace in many NLP applications such as sentiment classification (Chen et al., 2016; Yang et al., 2016), aspect-level sentiment analysis (Tay et al., 2018s, 2017b; Chen et al., 2017) and entailment classification (Rocktäschel et al., 2015). Co-attention / Bi-Attention (Xiong et al., 2016; Seo et al., 2016) is a form of pairwise attention mechanism that was proposed to model query-document pairs. Intra-attention can be interpreted as a self-targetted co-attention and is seeing a lot promising results in many recent works (Vaswani et al., 2017; Parikh et al., 2016; Tay et al., 2017a; Shen et al., 2017). The key idea is to model a sequence against itself, learning to attend while capturing long term dependencies and word-word level interactions. To the best of our knowledge, our work is not only the first work that only applies intra-attention to sarcasm detection but also the first attention model for sarcasm detection.

3 Our Proposed Approach

In this section, we describe our proposed model. Figure 1 illustrates our overall model architecture.

3.1 Input Encoding Layer

Our model accepts a sequence of one-hot encoded vectors as an input. Each one-hot encoded vector corresponds to a single word in the vocabulary. In the input encoding layer, each one-hot vector is converted into a low-dimensional vector representation (word embedding). The word embeddings are parameterized by an embedding layer $\mathbf{W} \in \mathbb{R}^{n \times |V|}$. As such, the output of this layer is a sequence of word embeddings, i.e., $\{w_1, w_2, \cdots w_\ell\}$ where ℓ is a predefined maximum sequence length.

3.2 Multi-dimensional Intra-Attention

In this section, we describe our multi-dimensional intra-attention mechanism for sarcasm detection. We first begin by describing the standard single-dimensional intra-attention. The multi-dimensional adaptation will be introduced later in this section. The key idea behind this layer is to *look in-between*, i.e., modeling the semantics between each word in the input sequence. We first begin by modeling the relationship of each word pair in the input sequence. A simple way to achieve this is to use a linear[1] transformation layer to project the concatenation of each word embedding **pair** into a scalar score as follows:

$$s_{ij} = W_a([w_i; w_j]) + b_a \tag{1}$$

where $W_a \in \mathbb{R}^{2n \times 1}, b_a \in \mathbb{R}$ are the parameters of this layer. $[.;.]$ is the vector concatenation operator and s_{ij} is a scalar representing the affinity score between word pairs (w_i, w_j). We can easily observe that s is a symmetrical matrix of $\ell \times \ell$ dimensions. In order to learn attention vector a, we apply a row-wise max-pooling operator on matrix s.

$$a = softmax(\max_{row} s) \tag{2}$$

where $a \in \mathbb{R}^\ell$ is a vector representing the learned intra-attention weights. Then, the vector a is employed to learn weighted representation of $\{w_1, w_2 \cdots w_\ell\}$ as follows:

$$v_a = \sum_{i=1}^{\ell} w_i a_i \tag{3}$$

[1]Early experiments found that adding nonlinearity here may degrade performance.

where $v \in \mathbb{R}^n$ is the intra-attentive representation of the input sequence. While other choices of pooling operators may be also employed (e.g., mean-pooling over max-pooling), the choice of max-pooling is empirically motivated. Intuitively, this attention layer learns to pay attention based on a word's *largest* contribution to all words in the sequence. Since our objective is to highlight words that might contribute to the contrastive theories of sarcasm, a more discriminative pooling operator is desirable. Notably, we also mask values of s where $i = j$ such that we do not allow the relationship scores of a word with respect to itself to influence the overall attention weights.

Furthermore, our network can be considered as an *'inner'* adaptation of neural attention, modeling intra-sentence relationships between the raw word representations instead of representations that have been compositionally manipulated. This allows word-to-word similarity to be modeled *'as it is'* and not be influenced by composition. For example, when using the outputs of a compositional encoder (e.g., LSTM), matching words n and $n + 1$ might not be meaningful since they would be relatively similar in terms of semantic composition. For relatively short documents (such as tweets), it is also intuitive that attention typically focuses on the last hidden representation.

Intuitively, the relationships between two words is often not straightforward. Words are complex and often hold more than one meanings (or word senses). As such, it might be beneficial to model *multiple views* between two words. This can be modeled by representing the word pair interaction with a vector instead of a scalar. As such, we propose a multi-dimensional adaptation of the intra-attention mechanism. The key idea here is that each word pair is projected down to a low-dimensional vector before we compute the affinity score, which allows it to not only capture one view (one scalar) but also multiple views. A modification to Equation (1) constitutes our Multi-Dimensional Intra-Attention variant.

$$s_{ij} = W_p(ReLU(W_q([w_i; w_j]) + b_q)) + b_p \quad (4)$$

where $W_q \in \mathbb{R}^{n \times k}, W_p \in \mathbb{R}^{k \times 1}, b_q \in \mathbb{R}^k, b_p \in \mathbb{R}$ are the parameters of this layer. The final intra-attentive representation is then learned with Equation (2) and Equation (3) which we do not repeat here for the sake of brevity.

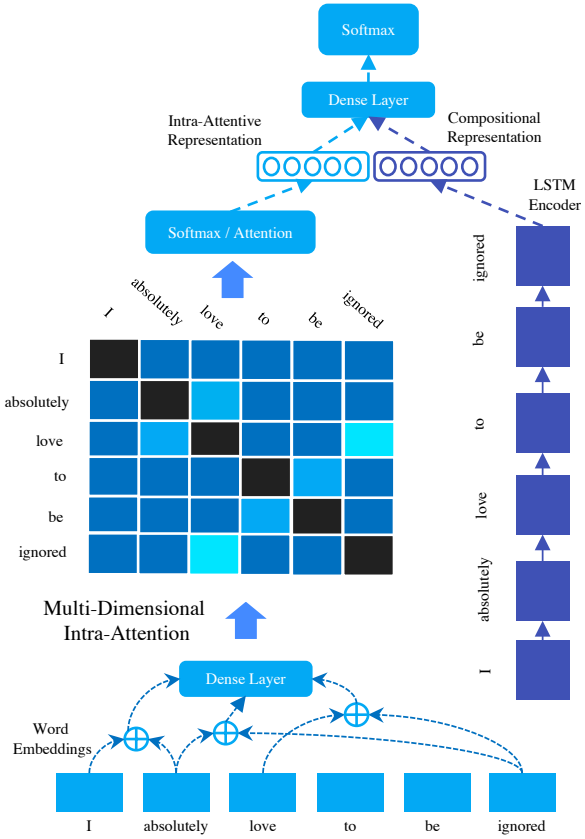

Figure 1: High level overview of our proposed MIARN architecture. MIARN learns two representations, one based on intra-sentence relationships (intra-attentive) and another based on sequential composition (LSTM). Both views are used for prediction.

3.3 Long Short-Term Memory Encoder

While we are able to simply use the learned representation v for prediction, it is clear that v does not encode compositional information and may miss out on important compositional phrases such as *'not happy'*. Clearly, our intra-attention mechanism simply considers a word-by-word interaction and does not model the input document sequentially. As such, it is beneficial to use a separate compositional encoder for this purpose, i.e., learning compositional representations. To this end, we employ the standard Long Short-Term Memory (LSTM) encoder. The output of an LSTM encoder at each time-step can be briefly defined as:

$$h_i = \text{LSTM}(w, i), \quad \forall i \in [1, \dots \ell] \quad (5)$$

where ℓ represents the maximum length of the sequence and $h_i \in \mathbb{R}^d$ is the hidden output of the LSTM encoder at time-step i. d is the size of the hidden units of the LSTM encoder. LSTM encoders are parameterized by gating mechanisms learned via nonlinear transformations. Since

LSTMs are commonplace in standard NLP applications, we omit the technical details for the sake of brevity. Finally, to obtain a compositional representation of the input document, we use $v_c = h_\ell$ which is the last hidden output of the LSTM encoder. Note that the inputs to the LSTM encoder are the word embeddings right after the input encoding layer and not the output of the intra-attention layer. We found that applying an LSTM on the intra-attentively scaled representations do not yield any benefits.

3.4 Prediction Layer

The inputs to the final prediction layer are two representations, namely (1) the intra-attentive representation ($v_a \in \mathbb{R}^n$) and (2) the compositional representation ($v_c \in \mathbb{R}^d$). This layer learns a joint representation of these two views using a nonlinear projection layer.

$$v = ReLU(W_z([v_a; v_c]) + b_z) \quad (6)$$

where $W_z \in \mathbb{R}^{(d+n)\times d}$ and $b_z \in \mathbb{R}^d$. Finally, we pass v into a Softmax classification layer.

$$\hat{y} = Softmax(W_f\, v + b_f) \quad (7)$$

where $W_f \in \mathbb{R}^{d\times 2}, b_f \in \mathbb{R}^2$ are the parameters of this layer. $\hat{y} \in \mathbb{R}^2$ is the output layer of our proposed model.

3.5 Optimization and Learning

Our network is trained end-to-end, optimizing the standard binary cross-entropy loss function.

$$J = -\sum_{i=1}^{N} [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] + R \quad (8)$$

where J is the cost function, $\hat{y}$ is the output of the network, $R = ||\theta||_{L2}$ is the L2 regularization and λ is the weight of the regularizer.

4 Empirical Evaluation

In this section, we describe our experimental setup and results. Our experiments were designed to answer the following research questions (**RQ**s).

- **RQ1** - Does our proposed approach outperform existing state-of-the-art models?
- **RQ2** - What are the impacts of some of the architectural choices of our model? How much does intra-attention contribute to the model performance? Is the Multi-Dimensional adaptation better than the Single-Dimensional adaptation?
- **RQ3** - What can we interpret from the intra-attention layers? Does this align with our hypothesis about *looking in-between* and modeling contrast?

4.1 Datasets

We conduct our experiments on six publicly available benchmark datasets which span across three well-known sources.

- **Tweets** - Twitter[2] is a microblogging platform which allows users to post statuses of less than 140 characters. We use two collections for sarcasm detection on tweets. More specifically, we use the dataset obtained from (1) (Ptáček et al., 2014) in which tweets are trained via hashtag based semi-supervised learning, i.e., hashtags such as #not, #sarcasm and #irony are marked as sarcastic tweets and (2) (Riloff et al., 2013) in which Tweets are hand annotated and manually checked for sarcasm. For both datasets, we retrieve. Tweets using the Twitter API using the provided tweet IDs.
- **Reddit** - Reddit[3] is a highly popular social forum and community. Similar to Tweets, sarcastic posts are obtained via the tag '/s' which are marked by the authors themselves. We use two Reddit datasets which are obtained from the subreddits */r/movies* and */r/technology* respectively. Datasets are subsets from (Khodak et al., 2017).
- **Debates** - We use two datasets[4] from the Internet Argument Corpus (IAC) (Lukin and Walker, 2017) which have been hand annotated for sarcasm. This dataset, unlike the first two, is mainly concerned with long text and provides a diverse comparison from the other datasets. The IAC corpus was designed for research on political debates on online forums. We use the V1 and V2 versions of the sarcasm corpus which are denoted as IAC-V1 and IAC-V2 respectively.

The statistics of the datasets used in our experiments is reported in Table 1.

[2] https://twitter.com
[3] https://reddit.com
[4] https://nlds.soe.ucsc.edu/sarcasm1

Dataset	Train	Dev	Test	Avg ℓ
Tweets (Ptáček et al.)	44017	5521	5467	18
Tweets (Riloff et al.)	1369	195	390	14
Reddit (/r/movies)	5895	655	1638	12
Reddit (/r/technology)	16146	1793	4571	11
Debates IAC-V1	3716	464	466	54
Debates IAC-V2	1549	193	193	64

Table 1: Statistics of datasets used in our experiments.

4.2 Compared Methods

We compare our proposed model with the following algorithms.

- **NBOW** is a simple neural bag-of-words baseline that sums all the word embeddings and passes the summed vector into a simple logistic regression layer.
- **CNN** is a vanilla Convolutional Neural Network with max-pooling. CNNs are considered as compositional encoders that capture n-gram features by parameterized sliding windows. The filter width is 3 and number of filters $f = 100$.
- **LSTM** is a vanilla Long Short-Term Memory Network. The size of the LSTM cell is set to $d = 100$.
- **ATT-LSTM (Attention-based LSTM)** is a LSTM model with a neural attention mechanism applied to all the LSTM hidden outputs. We use a similar adaptation to (Yang et al., 2016), albeit only at the document-level.
- **GRNN (Gated Recurrent Neural Network)** is a Bidirectional Gated Recurrent Unit (GRU) model that was proposed for sarcasm detection by (Zhang et al., 2016). GRNN uses a gated pooling mechanism to aggregate the hidden representations from a standard BiGRU model. Since we only compare on document-level sarcasm detection, we do not use the variant of GRNN that exploits user context.
- **CNN-LSTM-DNN (Convolutional LSTM + Deep Neural Network)**, proposed by (Ghosh and Veale, 2016), is the state-of-the-art model for sarcasm detection. This model is a combination of a CNN, LSTM and Deep Neural Network via stacking. It stacks two layers of 1D convolution with 2 LSTM layers. The output passes through a deep neural network (DNN) for prediction.

Both CNN-LSTM-DNN (Ghosh and Veale, 2016) and GRNN (Zhang et al., 2016) are state-of-the-art models for document-level sarcasm detection and have outperformed numerous neural and non-neural baselines. In particular, both works have well surpassed feature-based models (Support Vector Machines, etc.), as such we omit comparisons for the sake of brevity and focus comparisons with recent neural models instead. Moreover, since our work focuses only on document-level sarcasm detection, we do not compare against models that use external information such as user profiles, context, personality information (Ghosh and Veale, 2017) or emoji-based distant supervision (Felbo et al., 2017).

For our model, we report results on both multi-dimensional and single-dimensional intra-attention. The two models are named as MIARN and SIARN respectively.

4.3 Implementation Details and Metrics

We adopt standard the evaluation metrics for the sarcasm detection task, i.e., macro-averaged F1 and accuracy score. Additionally, we also report precision and recall scores. All deep learning models are implemented using TensorFlow (Abadi et al., 2015) and optimized on a NVIDIA GTX1070 GPU. Text is preprocessed with NLTK[5]'s Tweet tokenizer. Words that only appear once in the entire corpus are removed and marked with the UNK token. Document lengths are truncated at $40, 20, 80$ tokens for Twitter, Reddit and Debates dataset respectively. Mentions of other users on the Twitter dataset are replaced by '@USER'. Documents with URLs (i.e., containing 'http') are removed from the corpus. Documents with less than 5 tokens are also removed. The learning optimizer used is the RMSProp with an initial learning rate of 0.001. The L2 regularization is set to 10^{-8}. We initialize the word embedding layer with GloVe (Pennington et al., 2014). We use the GloVe model trained on 2B Tweets for the Tweets and Reddit dataset. The Glove model trained on Common Crawl is used for the Debates corpus. The size of the word embeddings is fixed at $d = 100$ and are fine-tuned during training. In all experiments, we use a development set to select the best hyperparameters. Each model is trained for a total of 30 epochs and the model is saved each time the performance

[5] https://nltk.org

	Tweets (Ptáček et al., 2014)				Tweets (Riloff et al., 2013)			
Model	P	R	F1	Acc	P	R	F1	Acc
NBOW	80.02	79.06	79.43	80.39	*71.28*	62.37	64.13	79.23
Vanilla CNN	82.13	79.67	80.39	81.65	71.04	67.13	68.55	*79.48*
Vanilla LSTM	*84.62*	83.21	83.67	*84.50*	67.33	67.20	67.27	76.27
Attention LSTM	84.16	*85.10*	83.67	84.40	68.78	*68.63*	*68.71*	77.69
GRNN (Zhang et al.)	84.06	83.02	83.43	84.20	66.32	64.74	65.40	76.41
CNN-LSTM-DNN (Ghosh and Veale)	84.06	83.45	*83.74*	84.39	69.76	66.62	67.81	78.72
SIARN (this paper)	85.02	84.27	84.59	85.24	**73.82**	**73.26**	**73.24**	**82.31**
MIARN (this paper)	**86.13**	**85.79**	**86.00**	**86.47**	73.34	68.34	70.10	80.77

Table 2: Experimental Results on Tweets datasets. Best result in is boldface and second best is underlined. Best performing baseline is in *italics*.

	Reddit (/r/movies)				Reddit (/r/technology)			
Model	P	R	F1	Acc	P	R	F1	Acc
NBOW	67.33	66.56	66.82	67.52	65.45	65.62	65.52	66.55
Vanilla CNN	65.97	65.97	65.97	66.24	65.88	62.90	62.85	66.80
Vanilla LSTM	67.57	67.67	67.32	67.34	66.94	67.22	67.03	*67.92*
Attention LSTM	68.11	67.87	67.94	68.37	*68.20*	*68.78*	*67.44*	67.22
GRNN (Zhang et al.)	66.16	66.16	66.16	66.42	66.56	66.73	66.66	67.65
CNN-LSTM-DNN (Ghosh and Veale)	*68.27*	*67.87*	*67.95*	*68.50*	66.14	66.73	65.74	66.00
SIARN (this paper)	69.59	**69.48**	69.52	69.84	**69.35**	**70.05**	**69.22**	69.57
MIARN (this paper)	**69.68**	69.37	**69.54**	**69.90**	68.97	69.30	69.09	**69.91**

Table 3: Experimental results on Reddit datasets. Best result in is boldface and second best is underlined. Best performing baseline is in *italics*.

	Debates (IAC-V1)				Debates (IAC-V2)			
Model	P	R	F1	Acc	P	R	F1	Acc
NBOW	57.17	57.03	57.00	57.51	66.01	66.03	66.02	66.09
Vanilla CNN	58.21	58.00	57.95	58.55	68.45	68.18	68.21	68.56
Vanilla LSTM	54.87	54.89	54.84	54.92	68.30	63.96	60.78	62.66
Attention LSTM	58.98	57.93	57.23	59.07	70.04	69.62	*69.63*	*69.96*
GRNN (Zhang et al.)	56.21	56.21	55.96	55.96	62.26	61.87	61.21	61.37
CNN-LSTM-DNN (Ghosh and Veale)	55.50	54.60	53.31	55.96	64.31	64.33	64.31	64.38
SIARN (this paper)	**63.94**	63.45	62.52	62.69	72.17	71.81	71.85	72.10
MIARN (this paper)	63.88	**63.71**	**63.18**	**63.21**	**72.92**	**72.93**	**72.75**	**72.75**

Table 4: Experimental results on Debates datasets. Best result in is boldface and second best is underlined. Best performing baseline is in *italics*.

on the development set is topped. The batch size is tuned amongst $\{128, 256, 512\}$ for all datasets. The only exception is the Tweets dataset from (Riloff et al., 2013), in which a batch size of 16 is used in lieu of the much smaller dataset size. For fair comparison, all models have the same hidden representation size and are set to 100 for both recurrent and convolutional based models (i.e., number of filters). For MIARN, the size of intra-attention hidden representation is tuned amongst $\{4, 8, 10, 20\}$.

4.4 Experimental Results

Table 2, Table 3 and Table 4 reports a performance comparison of all benchmarked models on the Tweets, Reddit and Debates datasets respectively. We observe that our proposed SIARN and MIARN models achieve the best results across all six datasets. The relative improvement differs across domain and datasets. On the Tweets dataset from (Ptáček et al., 2014), MIARN achieves about $\approx 2\% - 2.2\%$ improvement in terms of F1 and accuracy score when compared against the best baseline. On the other Tweets dataset from (Riloff et al., 2013), the performance gain of our proposed model is larger, i.e., $3\% - 5\%$ improvement on average over most baselines. Our proposed SIARN and MIARN models achieve very competitive performance on the Reddit datasets, with an average of $\approx 2\%$ margin improvement over the best baselines. Notably, the baselines we compare against are extremely competitive state-of-the-art neural network models. This further reinforces the effectiveness of our proposed approach. Additionally, the performance improvement on Debates (long text) is significantly larger than short

text (i.e., Twitter and Reddit). For example, MIARN outperforms GRNN and CNN-LSTM-DNN by $\approx 8\% - 10\%$ on both IAC-V1 and IAC-V2. At this note, we can safely put **RQ1** to rest.

Overall, the performance of MIARN is often marginally better than SIARN (with some exceptions, e.g., *Tweets* dataset from (Riloff et al., 2013)). We believe that this is attributed to the fact that more complex word-word relationships can be learned by using multi-dimensional values instead of single-dimensional scalars. The performance brought by our additional intra-attentive representations can be further observed by comparing against the vanilla LSTM model. Clearly, removing the intra-attention network reverts our model to the standard LSTM. The performance improvements are encouraging, leading to almost 10% improvement in terms of F1 and accuracy. On datasets with short text, the performance improvement is often a modest $\approx 2\% - 3\%$ (**RQ2**). Notably, our proposed models also perform much better on long text, which can be attributed to the intra-attentive representations explicitly modeling long range dependencies. Intuitively, this is problematic for models that only capture sequential dependencies (e.g., word by word).

Finally, the relative performance of competitor methods are as expected. NBOW performs the worse, since it is just a naive bag-of-words model without any compositional or sequential information. On short text, LSTMs are overall better than CNNs. However, this trend is reversed on long text (i.e., Debates) since the LSTM model may be overburdened by overly long sequences. On short text, we also found that attention (or the gated pooling mechanism from GRNN) did not really help make any significant improvements over the vanilla LSTM model and a qualitative explanation to why this is so is deferred to the next section. However, attention helps for long text (such as debates), resulting in Attention LSTMs becoming the strongest baseline on the Debates datasets. However, our proposed intra-attentive model is both effective on short text and long text, outperforming Attention LSTMs consistently on all datasets.

4.5 In-depth Model Analysis

In this section, we present an in-depth analysis of our proposed model. More specifically, we not only aim to showcase the interpretability of our model but also explain how representations are formed. More specifically, we test our model (trained on Tweets dataset by (Ptáček et al., 2014)) on two examples. We extract the attention maps of three models, namely MIARN, Attention LSTM (ATT-LSTM) and applying Attention mechanism directly on the word embeddings without using a LSTM encoder (ATT-RAW). Table 5 shows the visualization of the attention maps.

Label	Model	Sentence
True	MIARN	I totally love being ignored !!
	ATT-LSTM	I totally love being ignored !!
	ATT-RAW	I totally love being ignored !!
False	MIARN	Being ignored sucks big time
	ATT-LSTM	Being ignored sucks big time
	ATT-RAW	Being ignored sucks big time

Table 5: Visualization of normalized attention weights on three different attention models (*Best viewed in color*). The intensity denotes the strength of the attention weight on the word.

In the first example (*true* label), we notice that the attention maps of MIARN are focusing on the words *'love'* and *'ignored'*. This is in concert with our intuition about modeling contrast and incongruity. On the other hand, both ATT-LSTM and ATT-RAW learn very different attention maps. As for ATT-LSTM, the attention weight is focused completely on the last representation - the token *'!!'*. Additionally, we also observed that this is true for many examples in the Tweets and Reddit dataset. We believe that this is the reason why standard neural attention does not help as what the attention mechanism is learning is to select the last representation (i.e., vanilla LSTM). Without the LSTM encoder, the attention weights focus on *'love'* but not *'ignored'*. This fails to capture any concept of contrast or incongruity.

Next, we consider the *false* labeled example. This time, the attention maps of MIARN are not as distinct as before. However, they focus on sentiment-bearing words, composing the words *'ignored sucks'* to form the majority of the intra-attentive representation. This time, passing the vector made up of *'ignored sucks'* allows the subsequent layers to recognize that there is no contrasting situation or sentiment. Similarly, ATT-LSTM focuses on the last word *time* which is totally non-interpretable. On the other hand, ATT-RAW focuses on relatively non-meaningful words such as *'big'*.

Overall, we analyzed two cases (positive and negative labels) and found that MIARN produces

very explainable attention maps. In general, we found that MIARN is able to identify contrast and incongruity in sentences, allowing our model to better detect sarcasm. This is facilitated by modeling intra-sentence relationships. Notably, the standard vanilla attention is not explainable or interpretable.

5 Conclusion

Based on the intuition of intra-sentence similarity (i.e., *looking in-between*), we proposed a new neural network architecture for sarcasm detection. Our network incorporates a multi-dimensional intra-attention component that learns an intra-attentive representation of the sentence, enabling it to detect contrastive sentiment, situations and incongruity. Extensive experiments over six public benchmarks confirm the empirical effectiveness of our proposed model. Our proposed MIARN model outperforms strong state-of-the-art baselines such as GRNN and CNN-LSTM-DNN. Analysis of the intra-attention scores shows that our model learns highly interpretable attention weights, paving the way for more explainable neural sarcasm detection methods.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Silvio Amir, Byron C Wallace, Hao Lyu, and Paula Carvalho Mário J Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *arXiv preprint arXiv:1607.00976* .

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in twitter, a novel approach. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. pages 50–58.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1650–1659.

Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 452–461.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. *arXiv preprint arXiv:1708.00524* .

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@NAACL-HLT 2016, June 16, 2016, San Diego, California, USA*. pages 161–169. http://aclweb.org/anthology/W/W16/W16-0425.pdf.

Aniruddha Ghosh and Tony Veale. 2017. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. pages 482–491.

Rachel Giora. 1995. On irony and negation. *Discourse processes* 19(2):239–264.

Roberto González-Ibánez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*. Association for Computational Linguistics, pages 581–586.

Irazú Hernández-Farías, José-Miguel Benedí, and Paolo Rosso. 2015. Applying basic features from sentiment analysis for automatic irony detection. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, pages 337–344.

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

Aditya Joshi, Pushpak Bhattacharyya, and Mark J Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)* 50(5):73.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. volume 2, pages 757–762.

Aditya Joshi, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016. Are word embedding-based features useful for sarcasm detection? *arXiv preprint arXiv:1610.00883* .

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579* .

Stephanie Lukin and Marilyn Walker. 2017. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. *arXiv preprint arXiv:1708.08572* .

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .

Abhijit Mishra, Kuntal Dey, and Pushpak Bhattacharyya. 2017. Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. pages 377–387. https://doi.org/10.18653/v1/P17-1035.

Abhijit Mishra, Diptesh Kanojia, Seema Nagar, Kuntal Dey, and Pushpak Bhattacharyya. 2016. Harnessing cognitive features for sarcasm detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1104.pdf.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval* 2(1–2):1–135.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. pages 2249–2255.

Lotem Peled and Roi Reichart. 2017. Sarcasm SIGN: interpreting sarcasm with sentiment based monolingual machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. pages 1690–1700. https://doi.org/10.18653/v1/P17-1155.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. pages 213–223.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, pages 97–106.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation* 47(1):239–268.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 704–714. http://aclweb.org/anthology/D/D13/D13-1066.pdf.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* .

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* .

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. Disan: Directional self-attention network for rnn/cnn-free language understanding. *arXiv preprint arXiv:1709.04696* .

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018s. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *In Proceedings of the AAAI 2018, 5956-5963*.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017a. A compare-propagate architecture with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102* .

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017b. Dyadic memory networks for aspect-based sentiment analysis. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*. pages 107–116. https://doi.org/10.1145/3132847.3132936.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. pages 6000–6010.

Deirdre Wilson. 2006. The pragmatics of verbal irony: Echo or pretence? *Lingua* 116(10):1722–1743.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *CoRR* abs/1611.01604.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J Smola, and Eduard H Hovy. 2016. Hierarchical attention networks for document classification.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. pages 2449–2460. http://aclweb.org/anthology/C/C16/C16-1231.pdf.