Reasoning with Sarcasm by Reading In-between

Yi Tay[1], Luu Anh Tuan[2], Siu Cheung Hui[1] and Jian Su[2]

[1] Nanyang Technological University, Singapore

[2] Institute for Infocomm Research, Singapore

Background

- Sarcasm:
  - "*a form of verbal irony that is intended to express contempt or ridicule*" (The Free Dictionary)
  - commonly manifests on social communities (e.g. Twitter, Reddit)
- Prior work considered sarcasm to be a contrast between a positive and negative sentiment (Riloff et al., 2013)

"I love to be ignored!"

"Perfect movie for people who can't fall asleep"

- Scope of this work: sarcasm detection based on document's content and commonsense knowledge but not external knowledge, or user's profile and context

"I love to solve math problem everyday"

"Cool. It took me 10 hours to flight from Sydney to Melbourne."

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

A*STAR I²R

Motivation

- State-of-the-art sarcasm detection systems mainly rely on deep and sequential neural networks (Ghosh and Veale, 2016; Zhang et al., 2016):
    - compositional encoders (GRU, LSTM) are often employed, with the input document being parsed one word at a time
    - no explicit interaction between word pairs → hampers ability to explicitly model contrast, incongruity or juxtaposition of situations
    - difficult to capture long-range dependencies

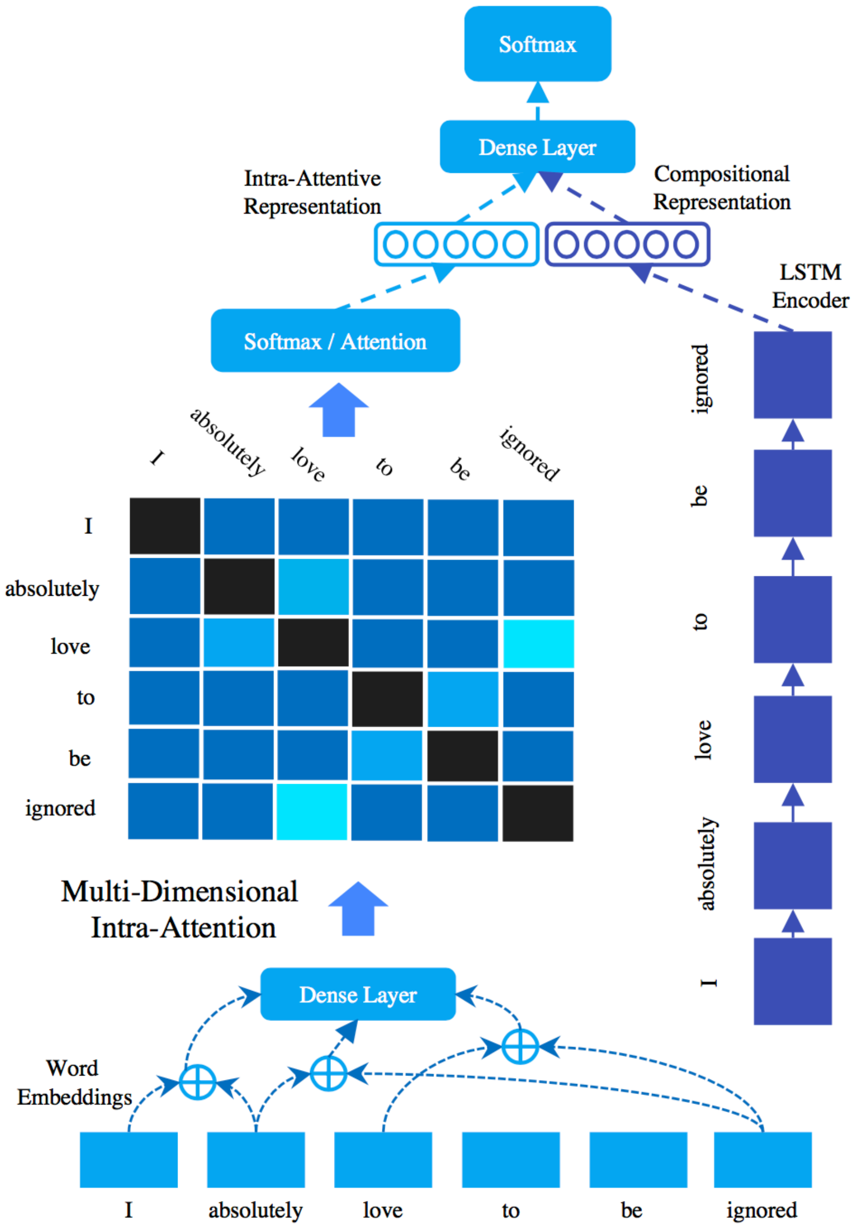
Proposed approach

- Our idea: modeling contrast in order to reason with sarcasm
    - either between positive-negative sentiments or between literal-figurative scenarios
- How?
    - looking in-between: propose a multi-dimensional intra-attention recurrent network → capture both word-word relationship and long-range dependency

"I absolutely love to be ignored!"

"Perfect movie for people who can't fall asleep"

Architecture



single-dimensional intra-attention:

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

multi-dimensional intra-attention:

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

intra-attention weight vector:

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

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

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

A*STAR I²R

Experiments

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

Experimental results

	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 (Ghosh and Veale)	84.06	83.45	*83.74*	84.39	69.76	66.62	67.81	78.72
SIARN (this paper)	<u>85.02</u>	<u>84.27</u>	<u>84.59</u>	<u>85.24</u>	**73.82**	**73.26**	**73.24**	**82.31**
MIARN (this paper)	**86.13**	**85.79**	**86.00**	**86.47**	<u>73.34</u>	<u>68.34</u>	<u>70.10</u>	<u>80.77</u>

Experimental results

	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 (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**

Experimental results

	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 (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**

Visualization of attention weights

Model	Sentence
MIARN	I totally love being ignored !!
ATT-LSTM	I totally love being ignored !!
ATT-RAW	I totally love being ignored !!

Conclusion

- We proposed a new neural network architecture for sarcasm detection
  - 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
- outperforms strong state-of-the-art baselines such as GRNN and CNN-LSTM-DNN over six public benchmarks
- Able to learns highly interpretable attention weights → paving the way for more explainable neural sarcasm detection methods.

References

[1] 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 EMNLP, 2013.

[2] Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. *Tweet sarcasm detection using deep neural network*. In proceedings of COLING, 2016.

[3] Aniruddha Ghosh and Tony Veale. 2016. *Fracking sarcasm using neural network*. In proceedings of NAACL, 2016.