The Role of Conversation Context for Sarcasm Detection in Online Interactions

Debanjan Ghosh[§] **Alexander Richard Fabbri**[†] **Smaranda Muresan**[‡]
[§]School of Communication Information, Rutgers University, NJ, USA
[†]Department of Computer Science, Columbia University, NY, USA
[‡]Data Science Institute, Columbia University, NY, USA
debanjan.ghosh@rutgers.edu, {arf2145,smara@columbia.edu}

Abstract

Computational models for sarcasm detection have often relied on the content of utterances in isolation. However, speaker's sarcastic intent is not always obvious without additional context. Focusing on social media discussions, we investigate two issues: (1) does modeling of conversation context help in sarcasm detection and (2) can we understand what part of conversation context triggered the sarcastic reply. To address the first issue, we investigate several types of Long Short-Term Memory (LSTM) networks that can model both the conversation context and the sarcastic response.[1] We show that the conditional LSTM network (Rocktäschel et al., 2015) and LSTM networks with sentence level attention on context and response outperform the LSTM model that reads only the response. To address the second issue, we present a qualitative analysis of attention weights produced by the LSTM models with attention and discuss the results compared with human performance on the task.

1 Introduction

It has been argued that sarcasm, or verbal irony, is a type of interactional phenomenon with specific perlocutionary effects on the hearer (Haverkate, 1990), such as to break their pattern of expectation. Thus, to be able to detect speakers' sarcastic intent it is necessary (even if maybe not sufficient) to consider their utterances in the larger conversation context. Consider the Twitter conversation example in Table 1. Without the context of UserA's statement, the sarcastic intent of UserB's response might not be detected.

Platform	Context-Reply pair
Twiter	**userA:** plane window shades are open . . . so that people can see if there is fire. **userB**: @UserA one more reason to feel really great.
Discussion Forum	**userC:** see for yourselves. The fact remains that in the caribbean, poverty and crime was near nil. Everyone was self-sufficient and contented with the standard of life. there were no huge social gaps. **userD:** Are you kidding me?! You think that Caribbean countries are "content?!" Maybe you should wander off the beach sometime and see for yourself.

Table 1: Sample Context/Reply pairs from two social media platforms

Most computational models for sarcasm detection have considered utterances in isolation (Davidov et al., 2010; González-Ibáñez et al., 2011; Liebrecht et al., 2013; Riloff et al., 2013; Maynard and Greenwood, 2014; Joshi et al., 2015; Ghosh et al., 2015; Joshi et al., 2016; Ghosh and Veale, 2016). In many instances, even humans have difficulty in recognizing sarcastic intent when considering an utterance in isolation (Wallace et al., 2014).

In this paper, we investigate the role of *conversation context* in detecting sarcasm in social media discussions (Twitter conversations and discussion forums). Table 1 shows some examples of sarcastic replies taken from two media platforms (userB

[1]We use response and reply interchangeably.

Proceedings of the SIGDIAL 2017 Conference, pages 186–196,
Saarbrücken, Germany, 15-17 August 2017.

and userD's posts, respectively) and a minimum unit of conversation context given by the prior turn (userA and userC's posts, respectively).

We address two specific issues: (1) does modeling of conversation context help in sarcasm detection and (2) can we understand what part of conversation context triggered the sarcastic reply (e.g., which sentence(s) from userC's comment triggered userD's sarcastic reply). To address the first issue, we investigate both SVM models with linguistically-motivated discrete features and several types of Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) that can model both the context and the sarcastic reply (Section 3). We show that the conditional LSTM network (Rocktäschel et al., 2015) and LSTM networks with sentence level attention on context and reply outperform the LSTM model that reads only the reply (Section 4). To address the second issue, we present a qualitative analysis of attention weights produced by the LSTM models with attention, and discuss the results compared with human performance on the task (Section 4.1). We make all datasets and code available.[2]

2 Data

One goal of our investigation is to comparatively study two types of social media platforms that have been considered individually for sarcasm detection: discussion forums and Twitter. We first discuss the two datasets and then point out some differences between them that could impact results and modeling choices.

Discussion Forums. Oraby et al. (2016) have introduced the Sarcasm Corpus V2, a subset of the Internet Argument Corpus that consists of discussion forum data. This corpus consists of sarcastic responses and their context (quotes to which the posts are replies to). The annotation of sarcastic vs. non-sarcastic replies was done using crowdsourcing, where annotators were asked to label a reply as sarcastic if any part of the reply contained sarcasm (thus annotation is done at the reply/comment level and not sentence level). The final gold sarcastic label was assigned only if a majority of the annotators labeled the reply as sarcastic. Although the dataset described by Oraby et al. (2016) consists of 9,400 post, only 50% (4,692 altogether; balanced between sarcastic and non-sarcastic categories) of that corpus is currently available for research.[3]

An example from this dataset is given in Table 1, where userD's reply has been labeled as sarcastic by annotators, in the context of userC's post/comment.

Twitter: To collect sarcastic and non-sarcastic tweets, we adopt the methodology proposed in related work (González-Ibáñez et al., 2011; Riloff et al., 2013; Bamman and Smith, 2015; Muresan et al., 2016). The sarcastic tweets were collected using hashtags such as, *#sarcasm*, *#sarcastic*, *#irony*, while the non-sarcastic tweets were the ones that do not contain these hashtags, but they might contain sentiment hashtags such as *#happy*, *#love*, *#sad*, *#hate*. We exclude the retweets, duplicates, quotes, tweets that contain only hashtags and URLs or are shorter than three words. Also, we eliminate all tweets where the hashtags of interest were not positioned at the very end of the message. Thus, we removed utterances such as "#sarcasm is something that I love". To built the conversation context, for each sarcastic and non-sarcastic utterance we used the "reply to status" parameter in the tweet to determine whether it was in reply to a previous tweet: if so, we downloaded the last tweet (i.e., "local conversation context") to which the original tweet was replying to (Bamman and Smith, 2015). In addition, we also collected the entire threaded conversation when available (Wang et al., 2015). Although we have collected over 200K tweets in the first step, around 13% of them were a reply to another tweet and thus our final Twitter conversations set contains 25,991 instances (12,215 instances for sarcastic class and 13,776 instances for the non-sarcastic class). We observe that 30% of the tweets have more than one tweet in the conversation context.

There are two main differences between these two datasets that need to be acknowledged. First, discussion forum posts are much longer than Twitter messages. Second, the way the gold labels for the sarcastic class are obtained is different. In the discussion forum dataset the gold label is obtained via crowdsourcing, thus the gold label emphasizes whether the sarcastic intent is *perceived* by hearers (we do not know if the speaker intended to be sarcastic or not). In Twitter dataset the gold label

[2] https://github.com/debanjanghosh/sarcasm_context

[3] This reduction in the training size will have obvious effects in the classification performance.

is given directly by the #hashtag the speaker used, signaling clearly the speaker's sarcastic intent. A third difference should be made: the size of the forum dataset is much smaller than the size of the Twitter dataset.

3 Computational Models and Experimental Setup

To assess the effect of conversation context (c) on labeling a reply (r) as sarcastic or not sarcastic, we consider two binary classification tasks. We refer to sarcastic instances as S and non-sarcastic instances as NS. In the first task, classification is performed using the reply in isolation (S^r vs. NS^r task). In the second, the classification considers both the reply and its context (S^{c+r} vs. NS^{c+r} task). We experiment with two types of computational models: Support Vector Machines (SVM) with linguistically-motivated discrete features (used as baseline; SVM$_{bl}$), and approaches using distributed representations. For the latter we use the Long short-term Memory (LSTM) Networks (Hochreiter and Schmidhuber, 1997) that have been shown to be successful in various NLP tasks, such as constituency parsing (Vinyals et al., 2015), language modeling (Zaremba et al., 2014), machine translation (Sutskever et al., 2014) and textual entailment (Bowman et al., 2015; Rocktäschel et al., 2015; Parikh et al., 2016). We present these models in the next subsections.

3.1 SVM with discrete features (SVM$_{bl}$)

For features, we used n-grams, lexicon-based features, and sarcasm indicators that are commonly used in the existing sarcasm detection approaches (Tchokni et al., 2014; González-Ibáñez et al., 2011; Riloff et al., 2013; Joshi et al., 2015; Ghosh et al., 2015; Muresan et al., 2016). Below is a short description of the features.

- **BoW:** Features are derived from unigram, bigram, and trigram representation of words.
- **Sentiment and Pragmatic features:** We use the Linguistic Inquiry and Word Count (LIWC) lexicon (Pennebaker et al., 2001) to identify the pragmatic features. Each category in this dictionary is treated as a separate feature and we define a Boolean feature that indicates if a context or a reply contains a LIWC category. Two sentiment lexicons are also used to model the utterance sentiment: "MPQA" (Wilson et al., 2005) and "Opinion Lexicon" (Hu and Liu, 2004). To capture sentiment, we count the number of positive and negative sentiment tokens, negations, and use a boolean feature that represents whether a reply contains both positive and negative sentiment tokens. For the S^{c+r} vs. NS^{c+r} classification task, we check whether the reply r has a different sentiment than the context c (similar to Joshi et al. (2015)). Given that sarcastic utterances often contain a positive sentiment towards a negative situation, we hypothesize that this feature will capture this type of sentiment incongruity.
- **Sarcasm Indicators:** Burgers et al. (2012) introduce a set of sarcasm indicators that explicitly signal if an utterance is sarcastic. We use *morpho-syntactic* features such as interjections (e.g., "uh", "oh", "yeah"), tag questions (e.g., "is not it?", "don't they"), exclamation marks (e.g., "!", "?"); *typographic* features such as capitalization of words, quotation marks, emoticons; *tropes* such as superlative and intensifiers words (e.g., "greatest", "best", "really") that often occur in sarcastic utterances (Camp, 2012).

When building the features, we lowercased the utterances, except the words where all the characters are uppercased (i.e., we did not lowercased "GREAT", "SO", and "WONDERFUL" in "GREAT i'm SO happy; shattered phone on this WONDERFUL day!!!"). Tokenization is conducted via CMU's Tweeboparser (Gimpel et al., 2011). For the discussion forum dataset we use the NLTK tool (Bird et al., 2009) for sentence boundary detection and tokenization. We used libSVM toolkit with Linear Kernel (Chang and Lin, 2011) with weights inversely proportional to the number of instances in each class.

3.2 Long Short-Term Memory Networks

LSTMs are a type of recurrent neural networks (RNNs) able to learn long-term dependencies (Hochreiter and Schmidhuber, 1997). Recently, LSTMs have been shown to be effective in Natural Language Inference (NLI) research, where the task is to establish the *relationship* between multiple inputs (i.e., a pair of premise and hypothesis as in the case of Recognizing Textual Entailment task (Bowman et al., 2015; Rocktäschel et al., 2015;

Parikh et al., 2016)). Since our goal is to explore the role of contextual information (our *first input*) for recognizing whether the reply (our *second input*) is sarcastic or not, we argue that using LSTM networks that read the context and reply are a natural modeling choice.

Attention-based LSTM Networks: Attentive neural networks have been shown to perform well on a variety of NLP tasks (Yang et al., 2016; Yin et al., 2015; Xu et al., 2015). Using attention-based LSTM will accomplish two goals: (1) test whether they achieve higher performance than simple LSTM models and (2) use the attention weights produced by the LSTM models to perform a qualitative analysis to determine which portions of context triggers the sarcastic reply.

Although Yang et al. (2016) have included two levels of attention mechanisms – one at the word level and another at the sentence level – we primarily focus on sentence level attention for two specific reasons. First, sentence level attentions can show the exact sentence in the context that is most informative to trigger sarcasm. In the discussion forum dataset, context posts are usually three or four sentences long and it could be helpful to identify the exact text that triggers the sarcastic reply. Second, attention over both the words and sentences seek to learn a large number of model parameters and given the moderate size of the discussion forum corpus they might overfit. For tweets, we treat each individual tweet as a sentence. The majority of tweets consist of a single sentence and even if there are multiple sentences in a tweet, often one sentence contains only hashtags, URLs, and emoticons making them uninformative if treated in isolation.

Figure 1 shows the high-level structure of the model. The context (left) is read by an LSTM ($LSTM_c$) whereas the response (right) is read by another LSTM ($LSTM_r$). We represent each sentence by the average of its word embeddings.

Let the context c contain d sentences and each sentence s_{c_i} contain T_{c_i} words. Similar to the notation of Yang et al. (2016), we first feed the sentence annotation h_{c_i} through a one layer MLP to get u_{c_i} as a hidden representation of h_{c_i}, then we weight the sentence u_{c_i} by measuring similarity with a sentence level context vector u_{c_s}. This gives a normalized importance weight α_{c_i} through a softmax function. v_c is the vector that summarize all the information of sentences in the context

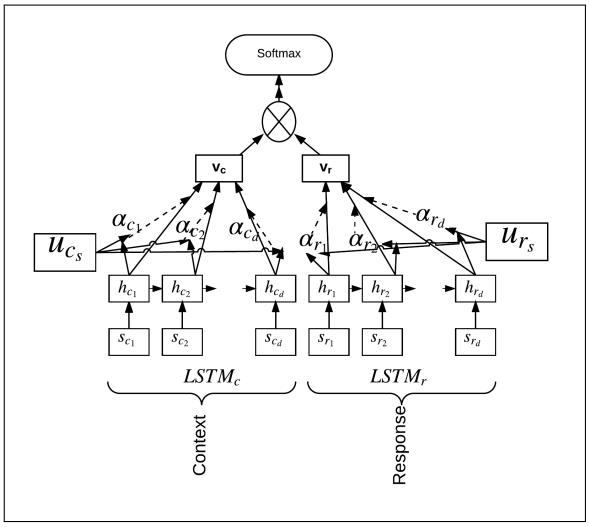

Figure 1: Sentence-level Attention Network for Context and Reply. Figure is inspired by Yang et al. (2016)

($LSTM_c$).

$$v_c = \sum_{i \in [1,d]} \alpha_{i_c} h_{i_c} \quad (1)$$

where attention is calculated as:

$$\alpha_{i_c} = \frac{\exp(u_{c_i}^T u_{c_s})}{\sum_{i \in [1,d]} \exp(u_{c_i}^T u_{c_s})} \quad (2)$$

Likewise we compute v_r for the response r via $LSTM_r$ (similar to eq. 1 and 2; also shown in Figure 1). Finally, we concatenate the vector v_c and v_r from the two LSTMs for the final softmax decision (i.e., predicting the S or NS class).

We also experiment with both word and sentence level attentions in a hierarchical fashion similarly to the approach proposed by Yang et al. (2016). As we show in Section 4 however, we achieve best performance for both datasets using just the sentence-level attention.

Conditional LSTM Networks: We also experiment with the *conditional encoding* model as introduced by Rocktäschel et al. (2015) for the task of recognizing textual entailment. In this architecture, two separate LSTMs are used – $LSTM_c$ and $LSTM_r$ – similar to the previous architecture without any attention, but for $LSTM_r$, its memory state is initialized with the last cell state of $LSTM_c$. In other words, $LSTM_r$ is conditioned on the representation of $LSTM_c$ that is built on the context.

Parameters and pre-trained word vectors. For both discussion forum and Twitter, we split randomly the corpus into training (80%), development (10%), and test (10%), maintaining the same distribution of sarcastic vs. non-sarcastic data in training, development and test. For Twitter we used the skip-gram word-embeddings (100-dimension) used in (Ghosh et al., 2015) that was built using over 2.5 million tweets.[4] For discussion forums, we use the standard Google n-gram $word2vec$ pre-trained model (300-dimension) (Mikolov et al., 2013). We do not optimize the word embedding during training. Out-of-vocabulary words in the training set are randomly initialized via sampling values uniformly from (-0.05,0.05). We use the development data to tune the parameters and selected dropout rate of 0.5 (from [.25,0.5, 0.75]), L_2 regularization strength and evaluate only that configuration on the test set. For both datasets mini-batch size of 16 is employed.

4 Results and Discussion

We report Precision (P), Recall (R), and F1 scores on S and NS classes. SVM$^{r}_{bl}$ and SVM$^{c+r}_{bl}$ respectively represent the performance of the SVM model using discrete features when using only the reply and the reply together with context. LSTM^{c_a} and LSTM^{r_a} are the attention-based LSTM models of context and reply, where the w, s and $w+s$ subscripts denote the word-level, sentence-level or word and sentence level attentions. LSTM^{conditional} is the *conditional encoding* model (no attention).

Discussion Forums: Table 2 shows the classification results on the discussion forum dataset. Although a vast majority of the context posts contain 3-4 sentences, around 100 context posts have more than ten sentences and thus we set a cutoff to a maximum of ten sentences for context modeling. For the reply r we considered the entire reply.

The SVM_{bl} models that are based on discrete features did not perform very well, and adding context actually hurt the performance. Regarding the performance of the neural network models, we observe that modeling context improves the performance using all types of LSTM architectures that read both context (c) and reply (r) (results are statistically significant when compared to LSTM^{r}). The highest performance when considering both the S and NS classes is achieved by the LSTM^{conditional} model (73.32% F1 for S class and 70.56% F1 for NS, showing a 6% and 3% improvement over LSTM^{r} for S and NS classes, respectively). The LSTM model with sentence-level attentions on both context and reply (LSTM$^{c_{a_s}}$+LSTM$^{r_{a_s}}$) gives the best F1 score of 73.7% for the S class. For the NS class, while we notice an improvement in precision we notice a drop in recall when compared to the LSTM model with sentence level attention only on reply (LSTM$^{r_{a_s}}$). Remember that sentence-level attentions are based on average word embeddings. We also experimented with the hierarchical attention model where each sentence is represented by a *weighted average* of its word embeddings. In this case, attentions are based on words and sentences and we follow the architecture of hierarchical attention network (Yang et al., 2016). We observe the performance (69.88% F1 for S category) deteriorates, probably due to the lack of enough training data. Since attention over both the words and sentences seek to learn a lot more model parameters, adding more training data will be helpful. With the full release of the Sarcasm Corpus used by Oraby et al. (2016), we expect to achieve better accuracy for these models.

Twitter: Table 3 shows the results on the Twitter dataset. As for discussion forums, adding context using the SVM models does not show a statistically significant improvement. For the neural networks model, similar to the results on discussion forums, the LSTM models that read both context and reply outperform the LSTM model that reads only the reply (LSTM^{r}). The best performing architectures are again the LSTM^{conditional} and LSTM with sentence-level attentions (LSTM$^{c_{a_s}}$+LSTM$^{r_{a_s}}$). LSTM^{conditional} model shows an improvement of 11% F1 on the S class and 4-5%F1 on the NS class, compared to LSTM^{r}. For the attention-based models, the improvement using context is smaller ($\sim$2% F1). We kept the maximum length of context to the last five tweets in the conversation context, when available.We also conducted experiments with only word-level attentions, however, we obtain lower accuracy in comparison to sentence level attention models.

[4] https://github.com/debanjanghosh/sarcasm_wsd

Experiment	S			NS		
	P	R	F1	P	R	F1
SVM$^{r}_{bl}$	65.55	66.67	66.10	66.10	64.96	65.52
SVM$^{c+r}_{bl}$	63.32	61.97	62.63	62.77	64.10	63.5
LSTM^{r}	67.90	66.23	67.1	67.08	**68.80**	67.93
LSTM^{c}+LSTM^{r}	66.19	79.49	72.23	74.33	59.40	66.03
LSTM^{conditional}	**70.03**	76.92	**73.32**	74.41	67.10	**70.56**
LSTM$^{r_{a_s}}$	69.45	70.94	70.19	70.30	68.80	69.45
LSTM$^{c_{a_s}}$+LSTM$^{r_{a_s}}$	66.90	**82.05**	**73.70**	**76.80**	59.40	66.99
LSTM$^{c_{a_{w+s}}}$+LSTM$^{r_{a_{w+s}}}$	65.90	74.35	69.88	70.59	61.53	65.75

Table 2: Experimental results for the discussion forum dataset (**bold** are best scores)

Experiment	S			NS		
	P	R	F1	P	R	F1
SVM$^{r}_{bl}$	64.20	64.95	64.57	69.0	68.30	68.7
SVM$^{c+r}_{bl}$	65.64	65.86	65.75	70.11	69.91	70.0
LSTM^{r}	73.25	58.72	65.19	61.47	75.44	67.74
LSTM^{c}+LSTM^{r}	70.89	67.95	69.39	64.94	68.03	66.45
LSTM^{conditional}	76.08	**76.53**	**76.30**	**72.93**	72.44	**72.68**
LSTM$^{r_{a_s}}$	76.00	73.18	74.56	70.52	73.52	71.9
LSTM$^{c_{a_s}}$+LSTM$^{r_{a_s}}$	**77.25**	75.51	**76.36**	72.65	**74.52**	**73.57**
LSTM$^{c_{a_w}}$+LSTM$^{r_{a_w}}$	76.74	69.77	73.09	68.63	75.77	72.02
LSTM$^{c_{a_{w+s}}}$+LSTM$^{r_{a_{w+s}}}$	76.42	71.37	73.81	69.50	74.77	72.04

Table 3: Experimental results for Twitter dataset (**bold** are best scores)

4.1 Qualitative Analysis

Wallace et al. (2014) showed that by providing contextual information humans are able to identify sarcastic utterances which they were unable without the context. However, it will be useful to understand whether a specific *part of the context* triggers the sarcastic reply.

To begin to address this issue, we conducted a qualitative study to understand whether (a) human annotators are able to identify parts of context that trigger the sarcastic reply and (b) attention weights are able to signal similar information. For (a) we designed a crowdsourcing experiment and for (b) we looked at the attention weights of the LSTM networks. Below is a short description of the crowdsourcing task.

4.1.1 Crowdsourcing Experiment.

We designed an Amazon Mechanical Turk task (for brevity, MTurk) framed as follow: Given a pair of context c and a sarcastic reply r from the discussion forum dataset, identify one or more sentences in c that may trigger the sarcastic reply r. Turkers could select one or more sentences from the context c, including the entire context. From the test data, we select examples with context length between three to seven sentences since for longer posts the task will be too complicated for the Turkers.

We provided a definition of sarcasm and a few examples to the Turkers. We also explained how to carry out the task with the help of a few context/reply pairs. Each HIT contains only one task and five Turkers were allowed to attempt each HIT (a total of 85 HITS). Turkers with reasonable quality (i.e., more than 95% of acceptance rate with experience of over 8,000 HITs) were selected and paid seven cents per task.

4.1.2 Comparing Turkers' answers with attention models.

We visualize and compare the sentence-level attention weights of the LSTM models on context with Turkers' annotations (Figure 2). We first measure the overlap of Turkers choice with the attention weights. For the sentence-based attention model (i.e., LSTM$^{c_{a_s}}$+LSTM$^{r_{a_s}}$ model for the discussion forum), we selected the sentence

with highest attention weight and matched it to the sentence selected by Turkers using majority voting. We found that 41% of times the sentence with the highest attention weight is also the one picked by Turkers. Figure 2 shows side by side the heat maps of the attention weights of LSTM models (LHS) and Turkers' choices when picking up sentences from context that they thought triggered the sarcastic reply (RHS).

Here the obvious question that we need to answer is why these sentences are selected by the models (and humans). In the next section we conduct a qualitative analysis to try answering this question.

4.1.3 Interpretation of selected context via attention weights

Semantic coherence between context and reply. Figure 2(a) depicts a case where the context contains three sentences and the attention weights given to the sentences are similar to the Turkers' choice. Looking at this example it seems the model pays attention to output vectors that are semantically coherent between c and r. The sarcastic response of this example contains a single sentence – "...hold your tongue ...in support of an anti-gay argument". The context contains the sentence S3 "...I've held my tongue on this as long as I can". The attention-based LSTM architecture is learning the attention weights simultaneously for the context c and the response r. Thus the model is showing contextual understanding by setting high weights to semantically coherent parts of the c and r. In Figure 2(b), attention weights is given to the most informative sentence –"rationally explain these creatures existence so recently in our human history if they were extinct for millions of years?". Here, the sarcastic reply mocks by claiming the author of the context is reading a lot more religious script (" you're reading waaaaay too much into your precious bible"). We also observe similar behavior in Tweets (highest attention to words *–retain* and *gerrymadering* in context: "breaking: *republicans retain majority* control of house" and reply: "hooray for *gerrymandering*" (Figure 3).

Incongruity between context and reply The meaning incongruity is an inherent characteristic of irony and sarcasm and have been extensively studied in linguistics, philosophy, communication science (Grice et al., 1975; Attardo, 2000; Burgers et al., 2012) as well as recently in NLP (Riloff et al., 2013; Joshi et al., 2015). For instance, Riloff et al. (2013) pointed out that identifying the incongruity between *positive* sentiment towards a *negative* situation is a key characteristic of sarcasm detection in social media. We observe in discussion forums and in Tweets that the attention-based models have frequently identified sentences and words from c and r that are semantically incongruous (i.e., opposite sentiment words). For instance, in Figure 2(c), the attention model has chosen sentence S1, which contains strong negative sentiment word ("disgusting sickening ..."). Interestingly, in contrast, the attention model on the reply, has given the highest weight to sentence that contain opposite sentiment ("I love you"). Thus, the model seems to learn the context incongruity of opposite sentiment for detecting sarcasm. However, it seems the Turkers prefer the second sentence S2 ("how can you tell a man that about his mum?") as the most instructive sentence instead of the first sentence. Looking at the sarcastic reply we observe that the reply contains remarks about "mothers" and apparently that commonality assisted the Turkers to chose the second sentence.

In Twitter dataset, we observe often the attention models have selected utterance(s) from the context which have opposite sentiment (Figure 4, Figure 5, and Figure 6). Here, the word and sentence-level attention model have chosen the particular utterance from the context (i.e., the top heatmap for the context) and the words with high attention (e.g., "mediocre", "gutsy").These words again show examples of meaning incongruity which is useful for sarcasm detection. Word-models seem to also work well when words in the context/reply are semantically incongruous but connected via deeper semantics ("bums" and "welfare" in context: "someone needs to remind these *bums* they work for the people" and reply: "feels like we are paying them *welfare*" (Figure 6).

Attention weights and sarcasm markers Looking just at attention weights in reply, we notice the models are giving highest weight to sentences that contain sarcasm markers, such as emoticons (i.e., ":p", ":)") and interjections (i.e., "ah", "hmm"). Sarcasm markers are explicit indicators of sarcasm that signal that an utterance is sarcastic, such as the use of emoticons, uppercase spelling of words, or interjections. (Attardo, 2000; Burgers et al., 2012). Use of such markers in

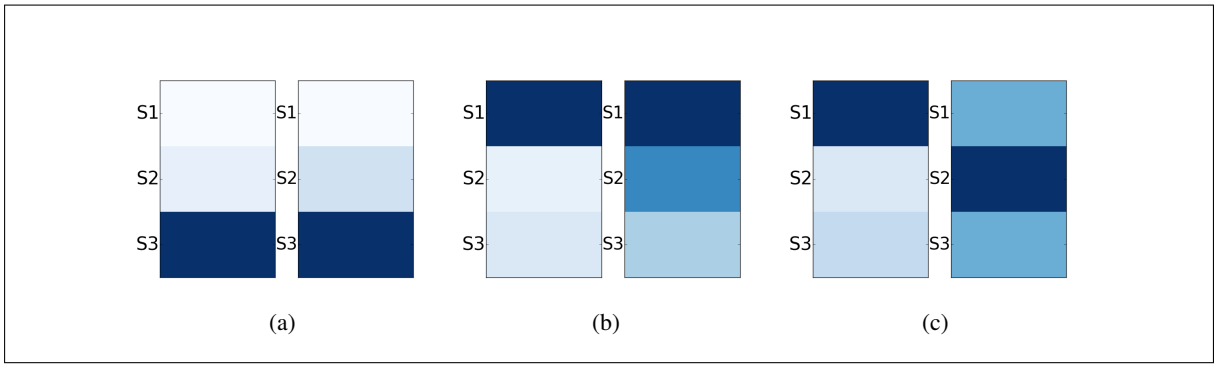

Figure 2: Context sentences that trigger sarcasm: LHS: *attention weights*; RHS: *Turkers' selections*

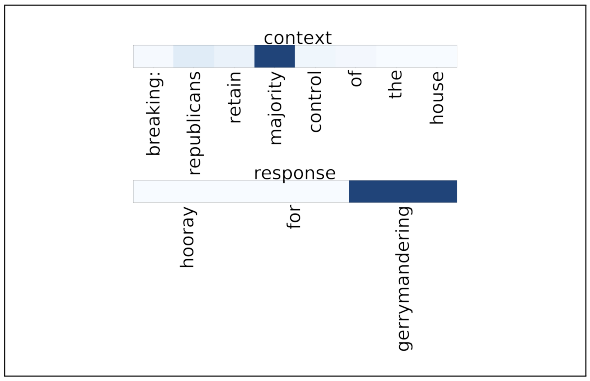

Figure 3: Attention visualization of semantic coherence between c and r

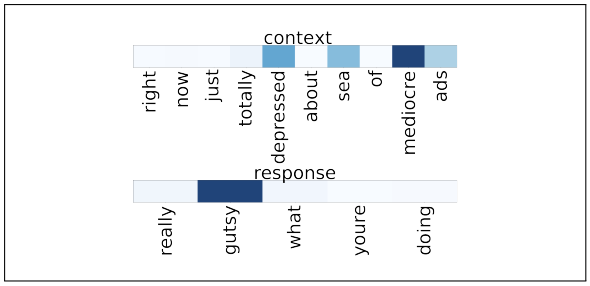

Figure 4: Attention visualization of incongruity between c and r

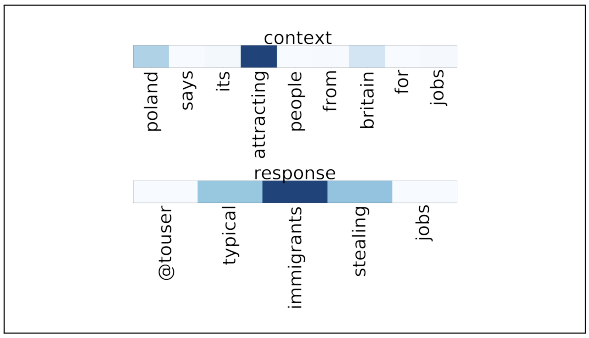

Figure 5: Attention visualization of incongruity between c and r

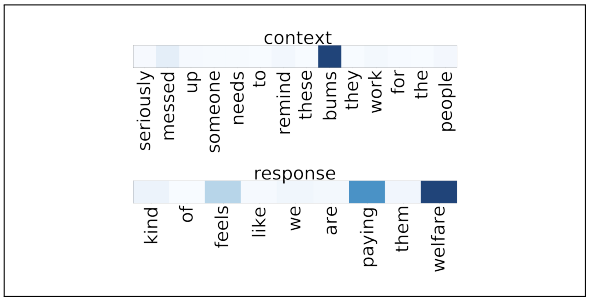

Figure 6: Attention visualization of incongruity between c and r

social media (particularly in Twitter) is extensive.

While we have started to understand the semantic of attention weights in this task, more studies need to be carry out. Rocktäschel et al. (2015) have argued that interpretations based on attentions weights have to be taken with care since the classification task is not forced to solely rely on the attentions weights. Thus in future work, we plan to analyze utterances that are more subtle and do not consist of sarcasm markers or explicit incongruence of opposite sentiment between context and response.

5 Related Work

Most computational models for sarcasm detection have considered utterances in isolation (Davidov et al., 2010; González-Ibáñez et al., 2011; Liebrecht et al., 2013; Riloff et al., 2013; Maynard and Greenwood, 2014; Ghosh et al., 2015; Joshi et al., 2016; Ghosh and Veale, 2016). However, even humans have difficulty sometimes in recognizing sarcastic intent when considering an utterance in isolation (Wallace et al., 2014). Thus, recent work on sarcasm and irony detection have started to exploit contextual information. In par-

ticular, (Khattri et al., 2015) analyzed authors' prior sentiment towards certain entities and if a new tweet deviates from the author's estimated sentiment the tweet is predicted to be sarcastic. Similar to this approach, several models have been introduced; some relied on extensive feature engineering to capture contextual information about authors, topics or conversation context whereas the rest are using deep learning techniques to embed authors' information (Rajadesingan et al., 2015). The two studies that have considered conversation context among other contextual information have shown minimal improvement when modeling conversation context using Twitter data (Bamman and Smith, 2015; Wang et al., 2015). Our work show that using better models, such as LSTM networks show a clear benefit of using context for sarcasm detection. As stated earlier in Section 3, LSTM's have been shown to be effective in NLI tasks, especially where the task is to establish the relationship between multiple inputs (i.e., in our case, between the context and the response). We observe that the LSTM^{conditional} model and the sentence level attention-based models using both context and reply present the best results.

6 Conclusion

This research makes a complementary contribution to existing work of modeling context for sarcasm/irony detection by looking at a particular type of context, *conversation context*. We have addressed two issues: (1) does modeling of conversation context help in sarcasm detection and (2) can we determine what part of the conversation context triggered the sarcastic reply. To answer the first question, we show that Long Short-Term Memory (LSTM) networks that can model both the context and the sarcastic reply achieve better performance than LSTM networks that read only the reply. In particular, conditional LSTM networks (Rocktäschel et al., 2015) and LSTM networks with sentence level attention achieved significant improvement (e.g., 6-11% F1 for discussion forums and Twitter messages). To address the second issue, we presented a qualitative analysis of attention weights produced by the LSTM models with attention, and discussed the results compared with human annotators. We also showed that attention-based models are able to identify inherent characteristics of sarcasm (i.e., sarcasm markers and sarcasm factors such as context incongruity). In future, we plan to study larger context, such as the full thread in a discussion forum that consider also the responses to the sarcastic comment, when available. We are also interested in analyzing sarcastic replies that do not contain sarcasm markers or explicit incongruence (i.e., opposing sentiment between the context and the reply).

Acknowledgements

This paper is based on work supported by the DARPA-DEFT program. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The authors thank Christopher Hidey for the discussions and resources on LSTM and the anonymous reviewers for helpful comments.

References

Salvatore Attardo. 2000. Irony markers and functions: Towards a goal-oriented theory of irony and its processing. *Rask* 12(1):3–20.

David Bamman and Noah A Smith. 2015. Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* .

Christian Burgers, Margot Van Mulken, and Peter Jan Schellens. 2012. Verbal irony differences in usage across written genres. *Journal of Language and Social Psychology* 31(3):290–310.

Elisabeth Camp. 2012. Sarcasm, pretense, and the semantics/pragmatics distinction*. *Noûs* 46(4):587–634.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. CoNLL '10.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of NAACL-HLT*. pages 161–169.

Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. 2015. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1003–1012.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the ACL*. pages 42–47.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *ACL (Short Papers)*. Association for Computational Linguistics, pages 581–586.

H Paul Grice, Peter Cole, and Jerry L Morgan. 1975. Syntax and semantics. *Logic and conversation* 3:41–58.

Henk Haverkate. 1990. A speech act analysis of irony. *Journal of Pragmatics* 14(1):77–109.

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

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pages 168–177.

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)*. Association for Computational Linguistics, Beijing, China, 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* .

Anupam Khattri, Aditya Joshi, Pushpak Bhattacharyya, and Mark James Carman. 2015. Your sentiment precedes you: Using an authors historical tweets to predict sarcasm. In *6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*. page 25.

CC Liebrecht, FA Kunneman, and APJ van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets# not .

Diana Maynard and Mark A Greenwood. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *Proceedings of LREC*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Smaranda Muresan, Roberto Gonzalez-Ibanez, Debanjan Ghosh, and Nina Wacholder. 2016. Identification of nonliteral language in social media: A case study on sarcasm. *Journal of the Association for Information Science and Technology* .

Shereen Oraby, Vrindavan Harrison, Ernesto Hernandez, Lena Reed, Ellen Riloff, and Marilyn Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue .

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933* .

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates* 71:2001.

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.

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 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 704–714.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Simo Tchokni, Diarmuid O Séaghdha, and Daniele Quercia. 2014. Emoticons and phrases: Status symbols in social media. In *Eighth International AAAI Conference on Weblogs and Social Media*.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*. pages 2773–2781.

Byron C Wallace, Laura Kertz Do Kook Choe, Laura Kertz, and Eugene Charniak. 2014. Humans require context to infer ironic intent (so computers probably do, too). In *ACL (2)*. pages 512–516.

Zelin Wang, Zhijian Wu, Ruimin Wang, and Yafeng Ren. 2015. Twitter sarcasm detection exploiting a context-based model. In *International Conference on Web Information Systems Engineering*. Springer, pages 77–91.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. Association for Computational Linguistics, pages 347–354.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. pages 2048–2057.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*. pages 1480–1489.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193* .

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329* .