Sarcasm Analysis Using Conversation Context

Debanjan Ghosh*
McGovern Institute for Brain Research
Massachusetts Institute of Technology
dg513@mit.edu

Alexander R. Fabbri
Department of Computer Science
Yale University
alexander.fabbri@yale.edu

Smaranda Muresan
Data Science Institute
Columbia University
smara@columbia.edu

Computational models for sarcasm detection have often relied on the content of utterances in isolation. However, the speaker's sarcastic intent is not always apparent without additional context. Focusing on social media discussions, we investigate three issues: (1) does modeling conversation context help in sarcasm detection? (2) can we identify what part of conversation context triggered the sarcastic reply? and (3) given a sarcastic post that contains multiple sentences, can we identify the specific sentence that is sarcastic? 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 current turn. We show that LSTM networks with sentence-level attention on context and current turn, as well as the conditional LSTM network, outperform the LSTM model that reads only the current turn. As conversation context, we consider the prior turn, the succeeding turn, or both. Our computational models are tested on two types of social media platforms: Twitter and discussion forums. We discuss several differences between these data sets, ranging from their size to the nature of the gold-label annotations. To address the latter two issues, we present a qualitative analysis of the attention weights produced by the LSTM models (with attention) and discuss the results compared with human performance on the two tasks.

1. Introduction

Social media has stimulated the production of user-generated content that contains figurative language use such as sarcasm and irony. Recognizing sarcasm and verbal

* The research was carried out while Debanjan was a Ph.D. candidate at Rutgers University.

Submission received: 15 October 2017; revised version received: 5 May 2018; accepted for publication: 20 August 2018.

doi:10.1162/coli_a_00336

irony is critical for understanding people's actual sentiments and beliefs (Maynard and Greenwood 2014). For instance, the utterance "I love waiting at the doctor's office for hours ..." is ironic, expressing a negative sentiment toward the situation of "waiting for hours at the doctor's office," even if the speaker uses positive sentiment words such as "love."

Verbal irony and sarcasm are a type of interactional phenomenon with specific perlocutionary effects on the hearer (Haverkate 1990), such as to break their pattern of expectation. For the current report, we do not make a clear distinction between sarcasm and verbal irony. Most computational models for sarcasm detection have considered utterances in isolation (Davidov, Tsur, and Rappoport 2010; González-Ibáñez, Muresan, and Wacholder 2011; Liebrecht, Kunneman, and Van den Bosch 2013; Riloff et al. 2013; Maynard and Greenwood 2014; Ghosh Guo, and Muresan 2015; Joshi, Sharma, and Bhattacharyya 2015; Ghosh and Veale 2016; Joshi et al. 2016b). In many instances, however, even humans have difficulty in recognizing sarcastic intent when considering an utterance in isolation (Wallace et al. 2014). Thus, to detect the speaker's sarcastic intent, it is necessary (even if maybe not sufficient) to consider their utterance(s) 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.

In this article, we investigate the role of *conversation context* for the detection of sarcasm in social media discussions (Twitter conversations and discussion forums). The unit of analysis (i.e., what we label as sarcastic or not sarcastic) is a message/turn in a social media conversation (i.e., a tweet in Twitter or a post/comment in discussion forums). We call this unit **current turn** (C_TURN). The conversation context that we consider is the **prior turn** (P_TURN), and, when available, also the **succeeding turn** (S_TURN), which is the reply to the current turn. Table 1 shows some examples of sarcastic messages (C_TURNs), together with their respective prior turns (P_TURN) taken from Twitter and two discussion forum corpora: the Internet Argument Corpus (IAC_{v2}) (Oraby et al. 2016) and Reddit (Khodak, Saunshi, and Vodrahalli 2018). Table 2 shows examples from the IAC_{v2} corpus of sarcastic messages (C_TURNs; userB's post) and the conversation context given by the prior turn (P_TURN; userA's post) as well as the succeeding turn (S_TURN; userC's post).

We address three specific questions:

1. Does modeling of conversation context help in sarcasm detection?
2. Can humans and computational models identify what part of the prior turn (P_TURN) triggered the sarcastic reply (C_TURN) (e.g., which sentence(s) from userC's turn triggered userD's sarcastic reply in Table 1)?
3. Given a sarcastic message (C_TURN) that contains multiple sentences, can humans and computational models identify the specific sentence that is sarcastic?

To answer the first question, we consider two types of context: (1) just the prior turn and (2) both the prior and the succeeding turns. We investigate both Support Vector Machine models (Cortes and Vapnik 1995; Chang and Lin 2011) with linguistically motivated discrete features and several types of Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber 1997) that can model both the conversation context (i.e., P_TURN, S_TURN or both) and the current turn

Table 1
Sarcastic turns (C_TURN) and their respective prior turns (P_TURN) in Twitter and discussion forums (Internet Argument Corpus, IAC) (Oraby et al. 2016) and Reddit.

Platform	Turn Type	Turn pairs
Twitter	P_TURN	**userA:** Plane window shades are open during take-off & landing so that people can see if there is fire in case of an accident URL.
	C_TURN	**userB**: @UserA ...awesome ...one more reason to feel really great about flying ...#sarcasm.
Discussion Forum (IAC_{v2})	P_TURN	**userC:** how do we rationally explain these creatures existence so recently in our human history if they were extinct for millions of years? and if they were the imaginings of bronze age sheep herders as your atheists/evolutionists would have you believe, then how did these ignorant people describe creatures we can now recognize from fossil evidence? and while your at it, ask yourself if it's reasonable that the bones of dead creatures have survived from 60 million years to some estimated to be more than 200 million years without becoming dust?
	C_TURN	**userD:** How about this explanation - you're reading WAAAAAY too much into your precious Bible.
Discussion Forum (*Reddit*)	P_TURN	**userE:** nothing will happen, this is going to die a quiet death like 99.99 % of other private member motions. this whole thing is being made into a big ordeal by those that either don't know how our parliament works, or are trying to push an agenda. feel free to let your mp know how you feel though but i doubt this motion gets more than a few minutes of discussion before it is send to the trashcan.
	C_TURN	**userF:** the usual "nothing to see here" response. whew! we can sleep at night and ignore this.

(C_TURN) (Section 4). We utilize different flavors of the LSTM networks and we show that the conditional LSTM network (Rocktäschel et al. 2016) and the LSTM networks with sentence-level attention on current turn (C_TURN) *and* context (particularly the prior turn) outperform the LSTM model, which reads only the current turn (C_TURN) (Section 5). We perform a detailed error analysis. Our computational models are tested on two different types of social media platforms: micro-blogging platforms such as Twitter and discussion forums. Our data sets, introduced in Section 3, differ on two main dimensions. First, discussion forum posts are much longer than Twitter messages, which makes them particularly relevant for the latter two questions we try to address. Second, the gold labels for the sarcastic class are obtained differently: Whereas Twitter and Reddit corpora are self-labeled (i.e., speakers themselves label their messages as sarcastic), the IAC_{v2} corpus is labeled via crowdsourcing. Thus, for the latter, the gold labels emphasize whether the sarcastic intent of the speaker has been *perceived* by the hearers/annotators (we do not know if the speaker intended to be sarcastic or not). We perform a study of training on Reddit data (self-labeled) and testing on IAC_{v2} (labeled via crowdsourcing). To answer the second and third questions, we present a qualitative analysis of the attention weights produced by the LSTM models with attention and

Table 2
Sarcastic messages (C_TURNs) and their respective prior turns (P_TURN) and succeeding turns (S_TURN) from IAC_{v2}.

Turn Type	Social media discussion
P_TURN	**userA:** my State is going to heII in a handbasket since these lefties took over. emoticonXBanghead.
C_TURN	**userB:** Well since Bush took office the mantra has been "Local Control" has it not. Apparently the people of your state want whats happening. Local control in action. Rejoice in your victory.
S_TURN	**userC:** I think the trip was a constructive idea, especially for high risk middle school youths .... Perhaps the program didn't respect their high risk homes enough. If it were a different group of students, the parents would have been told. The program was the YMCA, not lefty, but Christian based.
P_TURN	**userA:** In his early life, X had a reputation for drinking too much. Whether or not this affected his thinking is a question which should to be considered when asking questions about mormon theology ...emoticonXBanghead.
C_TURN	**userB:** Wow, that must be some good stuff he was drinking to keep him 'under the influence' for THAT long!! :p
S_TURN	**userC:** Perhaps he was stoned on other drugs like the early writers of the bible.

discuss the results compared with human performance on the tasks (Section 6). We make all data sets and code available.[1]

2. Related Work

Existing work in computational models for sarcasm detection addresses a variety of different tasks. These include, primarily, classifying sarcastic vs. non-sarcastic utterances using various lexical and pragmatic features (González-Ibáñez, Muresan, and Wacholder 2011; Liebrecht, Kunneman, and Van den Bosch 2013; Ghosh and Veale 2016; Joshi et al. 2016b; Muresan et al. 2016), rules and text-patterns (Veale and Hao 2010), specific hashtags (Maynard and Greenwood 2014) as well as semi-supervised approach (Davidov, Tsur, and Rappoport 2010). Researchers have also examined different characteristics of sarcasm, such as sarcasm detection as a sense-disambiguation problem (Ghosh, Guo, and Muresan 2015) and sarcasm as a contrast between a positive sentiment and negative situation (Riloff et al. 2013; Joshi, Sharma, and Bhattacharyya 2015). Apart from linguistically motivated contextual knowledge, cognitive features, such as eye-tracking information, are also used in sarcasm detection (Mishra et al. 2016). Schifanella et al. (2016) propose a multimodal approach, where textual and visual features are combined for sarcasm detection. Some studies present approaches for sarcasm detection in languages other than English. For example, Ptáček, Habernal, and

1 We use Theano Python library for the LSTM-based experiments. Code available at `https://github.com/debanjanghosh/sarcasm_context` and `https://github.com/Alex-Fabbri/deep_learning_nlp_sarcasm/`.

Hong (2014) use various *n*-grams, including unigrams, bigrams, and trigrams, and a set of language-independent features, such as punctuation marks, emoticons, quotes, capitalized words, and character *n*-gram features, to identify sarcasm in Czech tweets. Similarly, Liu et al. (2014) introduce POS sequences and homophony features to detect sarcasm from Chinese utterances. Bharti, Babu, and Jena (2017) compared tweets written in Hindi to news context for irony identification.

Most of these approaches have considered utterances in isolation. However, even humans have difficulty sometimes in recognizing sarcastic intent when considering an utterance in isolation (Wallace et al. 2014). Recently, an increasing number of researchers have started using contextual information for irony and sarcasm detection. The term context loosely refers to any *information* that is available beyond the utterance itself (Joshi, Bhattacharyya, and Carman 2017). There are two major research directions—*author* context and *conversation* context—and we briefly discuss them here.

Author Context. Researchers often examined the author-specific context (Khattri et al. 2015; Rajadesingan, Zafarani, and Liu 2015). For instance, Khattri et al. (2015) studied the historical tweets of a particular author to learn about the author's prevailing sentiment toward particular targets (e.g., named entities). Here, historical tweets are considered as the author's context. Khattri et al. hypothesized that altering sentiment toward a particular target in the candidate tweet may represent sarcasm. Rajadesingan, Zafarani, and Liu (2015) create features based on authors' previous tweets, for instance, an author's familiarity with sarcasm. Finally, Amir et al. (2016) enhanced Rajadesingan, Zafarani, and Liu's model by creating user embeddings based on the tweets of users and combined that with regular utterance-based embeddings for sarcasm detection.

Conversation Context. Wallace et al. (2014) present an annotation study where the annotators identify sarcastic comments from Reddit threads and were allowed to utilize additional context for sarcasm labeling. They also use a lexical classifier to automatically identify sarcastic comments and show that the model often fails to recognize the same examples for which the annotators requested more context. Bamman and Smith (2015) considered conversation context in addition to "author and addressee" features, which are derived from the author's historical tweets, profile information, and historical communication between the author and the addressee. Their results show only a minimal impact of modeling conversation context. Oraby et al. (2017) have studied the "pre" and "post" messages from debate forums as well as Twitter to identify whether rhetorical questions are used sarcastically or not. For both corpora, adding "pre" and "post" messages do not seem to significantly affect the F1 scores, even though using the "post" message as context seems to improve for the sarcastic class (Oraby et al. 2017). Unlike these approaches that model the utterance and context together, Wang et al. (2015) and Joshi et al. (2016a) use a sequence labeling approach and show that conversation helps in sarcasm detection. Inspired by this idea of modeling the current turn and context separately, in our prior work (Ghosh, Fabbri, and Muresan 2017)—which this paper substantially extends—we proposed a deep learning architecture based on LSTMs, where one LSTM reads the context (prior turn) and one LSTM reads the current turn, and showed that this type of architecture outperforms a simple LSTM that just reads the current turn. Independently, Ghosh and Veale (2017) have proposed a similar architecture based on Bi-LSTMs to detect sarcasm in Twitter. Unlike Ghosh and Veale, our prior work used attention-based LSTMs that allowed us to investigate whether we

can identify what part of the conversation context triggered the sarcastic reply, and showed results both on discussion forum data and Twitter.

This paper substantially extends our prior work introduced in Ghosh, Fabbri, and Muresan (2017). First, we extend the notion of context to consider also the "succeeding turn" with the "prior turn," and for that we collected a subcorpus from the Internet Argument Corpus (IAC) that contains both the prior turn and the succeeding turn as context. Second, we present a discussion on the nature of the data sets in terms of size and how the gold labels are obtained (self-labeled vs. crowdsource labeled), which might provide insights into the nature of sarcasm in social media. We use a new discussion forum data set from Reddit that is another example of a self-labeled data set (besides Twitter), where the speakers label their own post as sarcastic using the "/s" marker. We present an experiment where we train on the Reddit data set (self-labeled data) and test on IAC (where the gold labels were assigned via crowdsourcing). Third, we present a detailed error analysis of the computational models. Fourth, we address a new question: Given a sarcastic message that contains multiple sentences, can humans and computational models identify the specific sentence that is sarcastic? We conduct comparative analysis between human performance on the task and the attention weights of the LSTM models. In addition, for all the crowdsourcing experiments, we include more details on the interannotator agreement among Turkers. Fifth, we include new baselines (tf-idf; RBF kernels) and a run using unbalanced data sets. Last but not least, we empirically show that explicitly modeling the turns helps and provides better results than just concatenating the current turn and prior turn (and/or succeeding turn). This experimental result supports the conceptual claim that both we and Ghosh and Veale (2017) make that it is important to keep the C_TURN and the P_TURN (S_TURN) separate (e.g., modeled by different LSTMs), as the model is designed to recognize a possible inherent incongruity between them. This incongruity might become diffuse if the inputs are combined too soon (i.e., using one LSTM on combined current turn and context).

LSTM for Natural Language Inference (NLI) Tasks and Sarcasm Detection. LSTM networks are a particular type of recurrent neural networks that have been shown to be effective in NLI tasks, especially where the task is to establish the relationship between multiple inputs. For instance, in recognizing textual entailment research, LSTM networks, especially the attention-based models, are highly accurate (Bowman et al. 2015; Parikh et al. 2016; Rocktäschel et al. 2016; Sha et al. 2016). Rocktäschel et al. (2016) presented various word-based and conditional attention models that show how the entailment relationship between the *hypothesis* and the *premise* can be effectively derived. Parikh et al. (2016) use attention to decompose the RTE problem into subproblems that can be solved separately and Sha et al. (2016) presented an altered version ("re-read LSTM") of LSTM that is similar to word attention models of Rocktäschel et al. (2016). Likewise, recently LSTMs are used in sarcasm detection research (Ghosh and Veale 2017; Huang, Huang, and Chen 2017; Oraby et al. 2017). Oraby et al. (2017) used LSTM models to identify sarcastic utterances (tweets and posts from the IAC_{v2} that are structured as rhetorical questions), Huang, Huang, and Chen (2017) applied LSTM for sense-disambiguation research (on the same data set proposed by Ghosh, Guo, and Muresan [2015]), and Ghosh and Veale (2017) used bidirectional LSTMs to identify sarcastic tweets. In our research, we use multiple LSTMs for each text unit (e.g., the context and the response). We observe that the $\text{LSTM}^{conditional}$ model and the sentence-level attention-based models using both context and reply present the best results.

3. 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. In addition, the choice of our data sets reflects another critical aspect: the nature of the gold-label annotation of sarcastic messages. On the one hand, we have self-labeled data (i.e., the speakers themselves labeled their posts as sarcastic) in the case of Twitter and Reddit data. On the other hand, we have labels obtained via crowdsourcing as is the case for the IAC (Oraby et al. 2016). We first introduce the different data sets we use and then point out some differences between them that could impact results and modeling choices.

Internet Argument Corpus V2 (IAC_{v2}). Internet Argument Corpus (*IAC*) is a publicly available corpus of online forum conversations on a range of social and political topics, from gun control debates and marijuana legalization to climate change and evolution (Walker et al. 2012). The corpus comes with annotations of different types of pragmatic categories, such as agreement/disagreement (between a pair of online posts), nastiness, and sarcasm. There are different versions of *IAC* and we use a specific subset of *IAC* in this research. Oraby et al. (2016) have introduced a subset of the Internet Argument Corpus V2 that contains 9,400 posts labeled as sarcastic or non-sarcastic, called Sarcasm Corpus V2 (balanced data set). To obtain the gold labels, Oraby et al. (2016) first used a weakly supervised pattern learner to learn sarcastic and non-sarcastic patterns from the *IAC* posts and later utilized a multiple stage crowdsourcing process to identify sarcastic and non-sarcastic posts. Annotators were asked to label a post (current turn [C_TURN] in our terminology) as sarcastic if any part of the post contained sarcasm, and thus the annotation is done at the comment level and not the sentence level. This data set contains the post (C_TURN) as well as the quotes to which the posts are replies (i.e., prior turn [P_TURN] in our terminology). Sarcasm annotation was based on identifying three types of sarcasm: (a) general (i.e., mostly based on lexico-syntactic patterns); (b) rhetorical questions (i.e., questions that are not information-seeking questions but formed as an indirect assertion [Frank 1990], denoted as *RQ*); and (c) use of hyperbolic terms (use of "best," "greatest," "nicest," etc. [Camp 2012]). Although the data set described by Oraby et al. (2016) consists of 9,400 posts, only 50% of that corpus is currently available for research (4,692 altogether; balanced between sarcastic and non-sarcastic categories while maintaining the same distribution of general, hyperbolic, or *RQ* type sarcasm). This is the data set we used in our study and denote as IAC_{v2}.[2] Table 1 shows an example of sarcastic current turn (userD's post) and its prior turn (userC's post) from the IAC_{v2} data set.

The IAC_{v2} corpus contains only the prior turn as conversation context. Given that we are interested in studying also the succeeding turn as context, we checked to see whether for a current turn we can extract its succeeding turn from the general *IAC* corpus. Out of the 4,692 current turns, we found that a total of 2,309 have a succeeding turn. We denote this corpus as IAC^{+}_{v2}. Because a candidate turn can have more than one succeeding reply in the *IAC* corpus, the total size of the IAC^{+}_{v2} data set is 2,778. Examples from the IAC^{+}_{v2} are given in Table 2.

2 Oraby et al. (2016) reported best F1 scores between 65% and 74% for the three types of sarcasm. However, the reduction in the training size of the released corpus might negatively impact the classification performance.

Reddit Corpus. Khodak, Saunshi, and Vodrahalli (2018) introduce the Self-Annotated Reddit Corpus (SARC), which is a very large collection of sarcastic and non-sarcastic posts (over one million) from different subreddits. Similar to IAC_{v2}, this corpus also contains the prior turn as conversation context (the prior turn is either the original post or a prior turn in the discussion thread that the current turn is a reply to). Unlike IAC_{v2}, this corpus contains self-labeled data—that is, the speakers labeled their posts/comments as sarcastic using the marker "/s" at the end of sarcastic posts. For obvious reasons, the data are noisy because many users do not make use of the marker, do not know about it, or only use it where the sarcastic intent is not otherwise obvious. Khodak, Saunshi, and Vodrahalli have conducted an evaluation of the data, having three human evaluators manually check a random subset of 500 comments from the corpus tagged as sarcastic and 500 tagged as non-sarcastic, with full access to the post's context. They found around 3% of the non-sarcastic data is false negative. In their preliminary computational work on sarcasm detection, Khodak, Saunshi, and Vodrahalli have only selected posts between 2 and 50 words. For our research, we consider current turns that contain several sentences (between three to seven sentences). We selected a subset of the corpus (a total of 50K instances balanced between both the categories). We will refer to this corpus as the *Reddit* corpus. Table 1 shows an example of sarcastic current turn (userF's post) and its prior turn (userE's post) from the *Reddit* data set. We utilize standard preprocessing, such as sentence boundary detection and word tokenization, when necessary.[3]

Twitter Corpus. We have relied upon the annotations that users assign to their tweets using hashtags. We used Twitter developer APIs to collect tweets for our research.[4] The sarcastic tweets were collected using hashtags such as *#sarcasm*, *#sarcastic*, *#irony*. As non-sarcastic utterances, we consider sentiment tweets, that is, we adopt the methodology proposed in related work (González-Ibáñez, Muresan, and Wacholder 2011; Muresan et al. 2016). The non-sarcastic tweets were the ones that do not contain the sarcasm hashtags, but use hashtags that contain positive or negative sentiment words. The positive tweets express direct positive sentiment and they are collected based on tweets with positive hashtags such as *#happy*, *#love*, *#lucky*. Similarly, the negative tweets express direct negative sentiment and are collected based on tweets with negative hashtags such as *#sad*, *#hate*, *#angry*. Classifying sarcastic utterances against sentiment utterances is a considerably harder task than classifying against random objective tweets, since many sarcastic utterances also contain sentiment terms (González-Ibáñez, Muresan, and Wacholder 2011; Muresan et al. 2016). Table 3 shows all the hashtags used to collect the tweets. Similar to the *Reddit* corpus, this is a self-labeled data set, that is, the speakers use hashtags to label their posts as sarcastic. We exclude retweets (i.e., tweets that start with "RT"), duplicates, quotes, and tweets that contain only hashtags and URLs or are shorter than three words. We also eliminated tweets in languages other than English using the library Textblob.[5] 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."

3 Unless stated otherwise, we use the NLTK toolkit (Bird, Klein, and Loper 2009) for preprocessing.
4 Particularly, we use two libraries, the "twitter4j" (in Java) and the "twarc" (in Python) to accumulate the tweets.
5 Textblob: `http://textblob.readthedocs.io/en/dev/`.

Table 3
Hashtags for collecting sarcastic and non-sarcastic tweets.

Type	Hashtags
Sarcastic (S)	#sarcasm, #sarcastic, #irony
Non-Sarcastic (PosSent)	#happy, #joy, #happiness, #love, #grateful, #optimistic, #loved, #excited, #positive, #wonderful, #positivity, #lucky
Non-Sarcastic (NegSent)	#angry, #frustrated, #sad, #scared, #awful, #frustration, #disappointed, #fear, #sadness, #hate, #stressed

To build the conversation context, for each sarcastic and non-sarcastic tweet 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 (Bamman and Smith 2015). In addition, we also collected the entire threaded conversation when available (Wang et al. 2015). Although we 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 denote this data set as the *Twitter* data set. We notice that 30% of the tweets have more than one tweet in the conversation context. Table 1 shows an example of sarcastic current turn (userB's post) and its prior turn (userA's post) from the *Twitter* data set.

There are two main differences between these data sets 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. For the IAC_{v2} and IAC^{+}_{v2} data sets, the gold label is obtained via crowdsourcing; thus, the gold label emphasizes whether the sarcastic intent is *perceived* by the annotators (we do not know if the author intended to be sarcastic or not). For the *Twitter* and the *Reddit* data sets, the gold labels are given directly by the speakers (using hashtags on Twitter and the "/s" marker in Reddit), signaling clearly the speaker's sarcastic intent. A third difference should be noted: The size of the IAC_{v2} and IAC^{+}_{v2} data sets is much smaller than the size of the *Twitter* and *Reddit* data sets.

Table 4 presents the size of the training, development, and test data for the four corpora. Table 5 presents the average number of words per post and the number of average sentences per post. The average number of words/post for the two discussion forums are comparable.

Table 4
Data sets description (number of instances in Train/Dev/Test).

Corpus	Train	Dev	Test
Twitter	20,792	2,600	2,600
IAC_{v2}	3,756	468	468
IAC^{+}_{v2}	2,223	279	276
Reddit	40,000	5,000	5,000

Table 5
Average words/post and sentences/post from the three corpora.

Corpus	P_TURN		C_TURN		S_TURN	
	#words	#sents.	#words	#sents.	#words	#sents.
Twitter	17.48	1.71	16.92	1.51	-	-
IAC_{v2}	57.22	3.18	52.94	3.50	-	-
IAC^{+}_{v2}	47.94	2.55	42.82	2.98	43.48	3.04
Reddit	65.95	4.14	55.53	3.92	-	-

4. Computational Models and Experimental Set-up

To answer the first research question—does modeling of conversation context help in sarcasm detection—we consider two binary classification tasks. We refer to sarcastic instances as *S* and non-sarcastic instances as *NS*.

The first task is to predict whether the current turn (C_TURN abbreviated as *ct*) is sarcastic or not, considering it in isolation—S^{ct} vs. NS^{ct} task.

The second task is to predict whether the current turn is sarcastic or not, considering both the current turn and its conversation context given by the prior turn (P_TURN, abbreviated as *pt*), succeeding turn (S_TURN, abbreviated as *st*), or both—$S^{ct+context}$ vs. $NS^{ct+context}$ task, where *context* is *pt*, *st*, or *pt+st*.

For all the corpora introduced in Section 3—IAC_{v2}, IAC^{+}_{v2}, *Reddit*, and *Twitter*—we conduct S^{ct} vs. NS^{ct} and S^{ct+pt} vs. NS^{ct+pt} classification tasks. For IAC^{+}_{v2} we also perform experiments considering the succeeding turn *st* as conversation context (i.e., S^{ct+st} vs. NS^{ct+st} and $S^{ct+pt+st}$ vs. $NS^{ct+pt+st}$).

We experiment with two types of computational models: (1) support vector machines (SVM) (Cortes and Vapnik 1995; Chang and Lin 2011) with linguistically motivated discrete features (used as one baseline; disc_{bl}) and with tf-idf representations of the *n*-grams (used as another baseline; tf-idf_{bl}), and (2) approaches using distributed representations. For the latter, we use the LSTM networks (Hochreiter and Schmidhuber 1997), which have been shown to be successful in various NLP tasks, such as constituency parsing (Vinyals et al. 2015), language modeling (Zaremba, Sutskever, and Vinyals 2014), machine translation (Sutskever, Vinyals, and Le 2014), and textual entailment (Bowman et al. 2015; Parikh et al. 2016; Rocktäschel et al. 2016). We present these models in the next sections.

4.1 Baselines

For features, we used *n*-grams, lexicon-based features, and sarcasm indicators that are commonly used in the existing sarcasm detection approaches (González-Ibáñez, Muresan, and Wacholder 2011; Riloff et al. 2013; Tchokni, Séaghdha, and Quercia 2014; Ghosh, Guo, and Muresan 2015; Joshi, Sharma, and Bhattacharyya 2015; Muresan et al. 2016; Ghosh and Muresan 2018). We now provide a short description of the features.

BoW. Features are derived from unigram, bigram, and trigram representation of words.

Lexicon-Based Features. The lexicon-based features are derived from Pennebaker et al.'s Linguistic Inquiry and Word Count (LIWC) (Pennebaker, Francis, and Booth

2015) dictionary and emotion words from WordNet-Affect (Strapparava et al. 2004). The LIWC dictionary has been used widely in computational approaches to sarcasm detection (González-Ibáñez, Muresan, and Wacholder 2011; Justo et al. 2014; Muresan et al. 2016). It consists of a set of 64 word categories ranging from different *Linguistic Processes* (e.g., Adverbs, Past Tense, Negation), *Psychological Processes* (e.g., Positive Emotions, Negative Emotions, Perceptual Processes [See, Hear, Feel], SocialProcesses); *Personal Concerns* (e.g., Work, Achievement, Leisure); and *Spoken Categories* (Assent, Non-fluencies, Fillers). The LIWC dictionary contains around 4,500 words and word stems. 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 an LIWC category. WordNet-Affect (Strapparava et al. 2004) is an affective lexical resource of words that extends WordNet by assigning a variety of affect labels to a subset of synsets representing affective concepts in WordNet. Similarly to Muresan et al. (2016), we used the words annotated for associations with six emotions considered to be the most basic—joy, sadness, fear, disgust, anger, and surprise (Ekman 1992)—a total of 1,536 words.

Turn-Level Sentiment Features. Two sentiment lexicons are also used to model the turn sentiment: the MPQA Sentiment Lexicon (Wilson, Wiebe, and Hoffmann 2005), which contains over 8,000 positive, negative, and neutral sentiment words, and an opinion lexicon that contains around 6,800 positive and negative sentiment words (Hu and Liu 2004). To capture sentiment at turn level, we count the number of positive and negative sentiment tokens, negations, and use a Boolean feature that represents whether a turn contains both positive and negative sentiment tokens. For the S^{ct+pt} vs. NS^{ct+pt} classification task, we check whether the current turn *ct* has a different sentiment than the prior turn *pt* (similar to Joshi, Sharma, and Bhattacharyya [2015]). Given that sarcastic utterances often contain a positive sentiment toward a negative situation, we hypothesize that this feature will capture this type of sentiment incongruity.

Sarcasm Markers. Burgers, Van Mulken, and Schellens (2012) introduce a set of sarcasm markers that explicitly signal if an utterance is sarcastic. These markers are the metacommunicative clues that inform the reader that an utterance is sarcastic (Ghosh and Muresan 2018). Three types of markers—tropes (e.g., hyperbole), morpho-syntactic, and typographic—are used as features.

1. *Hyperbolic words:* Hyperboles or intensifiers are commonly used in sarcasm because speakers frequently overstate the magnitude of a situation or event. We use the MPQA lexicon (Wilson, Wiebe, and Hoffmann 2005) to select hyperbolic words, i.e., words with very strong subjectivity. These words (e.g., "greatest," "best," "nicest") are common in ironic and sarcastic utterances (Camp 2012).
2. *Morpho-syntactic:*
    - *Exclamations:* The use of exclamations ("!") is standard in expressions of irony and sarcasm. They emphasize a sense of surprise on the literal evaluation that is reversed in the ironic reading (Burgers 2010). Two binary features identify whether there is a single or multiple exclamation marks in the utterance.

- *Tag Questions:* As shown in Burgers (2010), tag questions are common in ironic utterances. We built a list of tag questions (e.g., "didn't you?", "aren't we?") from a grammar site and use them as binary indicators.[6]
- *Interjections:* Interjections seem to undermine a literal evaluation and occur frequently in ironic utterances (e.g., "'yeah," "wow," "yay," "ouch"). Similar to tag questions we drew interjections (a total of 250) from different grammar sites.

3. *Typography*:
   - *Capitalization:* Capitalization expresses excessive stress and thus it is standard in sarcastic posts on social media. For instance the words "GREAT," "SO," and "WONDERFUL" are indicators of sarcasm in the example "GREAT i'm SO happy; shattered phone on this WONDERFUL day!!!."
   - *Quotations:* This feature identifies whether any quotation appears in the utterance (i.e., replying to another message sarcastically).
   - *Emoticons:* Emoticons are frequently used to emphasize the sarcastic intent of the user. In the example "I love the weather ;) #sarcasm", the emoticon ";)" (wink) alerts the reader to a possible sarcastic interpretation. We collected a comprehensive list of emoticons (over 100) from Wikipedia and also used standard regular expressions to identify emoticons in our data sets.[7] Besides using the emoticons directly as binary features, we use their sentiment as features as well (e.g., "wink" is regarded as positive sentiment in MPQA).
   - *Punctuations:* Punctuation marks such as "?", ".", ";" and their various uses (e.g., single/multiple/mix of two different punctuations) are used as features.

When building the features, we lowercased the utterances, except the words where all the characters are uppercased (i.e., we did not lowercase "GREAT," "SO," and "WONDERFUL" in the example given). Twitter tokenization is done by CMU's Tweeboparser (Gimpel et al. 2011). For the discussion forum data set we use the NLTK tool (Bird, Klein, and Loper 2009) for sentence boundary detection and tokenization. We used the libSVM toolkit with Linear Kernel as well as RBF Kernel (Chang and Lin 2011) with weights inversely proportional to the number of instances in each class. The SVM models build with these discrete features will be one of our baselines (disc_{bl}).

We also computed another baseline based on the tf-idf (i.e., term-frequency-inverse-document-frequency) features of the n-grams (unigrams, bigrams, and trigrams) from the respective turns and used SVM for the classification. The count of a candidate n-gram in a turn is the term-frequency. The inverse document frequency is the logarithm of the division between total number of turns and number of turns with the n-gram in the training data set. This baseline is represented as tf-idf_{bl} in the following sections.

6 http://www.perfect-english-grammar.com/tag-questions.html.
7 http://sentiment.christopherpotts.net/code-data/.

4.2 Long Short-Term Memory Networks

LSTMs are a type of recurrent neural networks able to learn long-term dependencies (Hochreiter and Schmidhuber 1997). Recently, LSTMs have been shown to be effective in NLI tasks such as Recognizing Textual Entailment, where the goal is to establish the *relationship* between two inputs (e.g., a premise and a hypothesis) (Bowman et al. 2015; Parikh et al. 2016; Rocktäschel et al. 2016). LSTMs address the vanishing gradient problem commonly found in recurrent neural networks by incorporating gating functions into their state dynamics (Hochreiter and Schmidhuber 1997). We introduce some notations and terminology standard in the LSTM literature (Tai, Socher, and Manning 2015). The LSTM unit at each time step t is defined as a collection of vectors: an input gate i_t, a forget gate f_t, an output gate o_t, a memory cell c_t, and a hidden state h_t. The LSTM transition equations are:

$$\begin{aligned} i_t &= \sigma(\mathbf{W}_i * [h_{t-1}, x_t] + b_i) \\ f_t &= \sigma(\mathbf{W}_f * [h_{t-1}, x_t] + b_f) \\ o_t &= \sigma(\mathbf{W}_o * [h_{t-1}, x_t] + b_o) \\ \tilde{C}_t &= \tanh(\mathbf{W}_c * [h_{t-1}, x_t] + b_c) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{C}_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{1}$$

where x_t is the input at the current time step, σ is the logistic sigmoid function, and $\odot$ denotes element-wise multiplication. The input gate controls how much each unit is updated, the forget gate controls the extent to which the previous memory cell is forgotten, and the output gate controls the exposure of the internal memory state. The hidden state vector is a gated, partial view of the state of the unit's internal memory cell. Because the value of the gating variables vary for each vector element, the model can learn to represent information over multiple time scales.

As our goal is to explore the role of contextual information (e.g., prior turn and/or succeeding turn) for recognizing whether the current turn is sarcastic or not, we will use *multiple LSTMs*: one that reads the current turn and one (or two) that read(s) the context (e.g., one LSTM will read the prior turn and one will read the succeeding turn when available).

Attention-based LSTM Networks. Attentive neural networks have been shown to perform well on a variety of NLP tasks (Xu et al. 2015; Yang et al. 2016; Yin et al. 2016). 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 the qualitative analyses that enable us to answer the latter two questions we want to address (e.g., which portions of context triggers the sarcastic reply).

Yang et al. (2016) have included two levels of attention mechanisms, one at the word level and another at the sentence level, where the sentences are in turn produced by attentions over words (i.e., the hierarchical model). We experiment with two architectures: one hierarchical that uses both word-level and sentence-level attention (Yang et al. 2016), and one that uses only sentence-level attention (here we use only the average word embeddings to represent the sentences). One question we want to address is

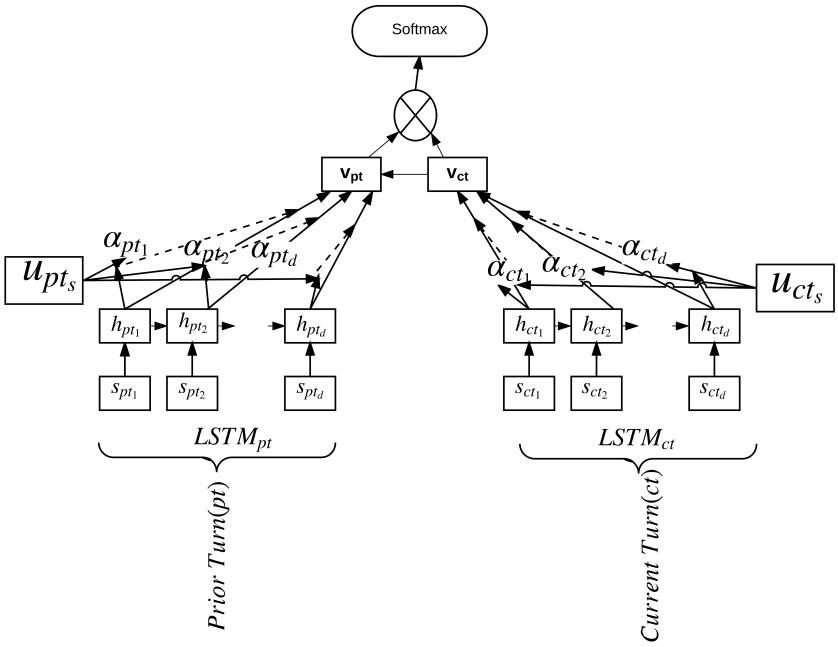


Figure 1
Sentence-level attention network for prior turn *pt* and current turn *ct*. Figure is inspired by Yang et al. (2016).

whether the sentence-level attention weights indicate what sentence(s) in the prior turn trigger(s) the sarcastic reply. In the discussion forum data sets, prior turns are usually more than three sentences long and thus the attention weights could indicate what part of the prior turn triggers the sarcastic post *ct*.

Figure 1 shows the high-level structure of the model, where the conversation context is represented by the prior turn *pt*. The context (left) is read by an LSTM ($LSTM_{pt}$) and the current turn *ct* (right) is read by another LSTM ($LSTM_{ct}$). Note that, for the model where we consider the succeeding turn *st* as well, we simply use another LSTM to read *st*. For brevity, we only show the sentence-level attention.

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

$$v_{pt} = \sum_{i \in [1,d]} \alpha_{pt_i} h_{pt_i} \tag{2}$$

where attention is calculated as:

$$\alpha_{pt_i} = \frac{\exp(u_{pt_i}^T u_{pt_s})}{\sum_{i \in [1,d]} \exp(u_{pt_i}^T u_{pt_s})} \tag{3}$$

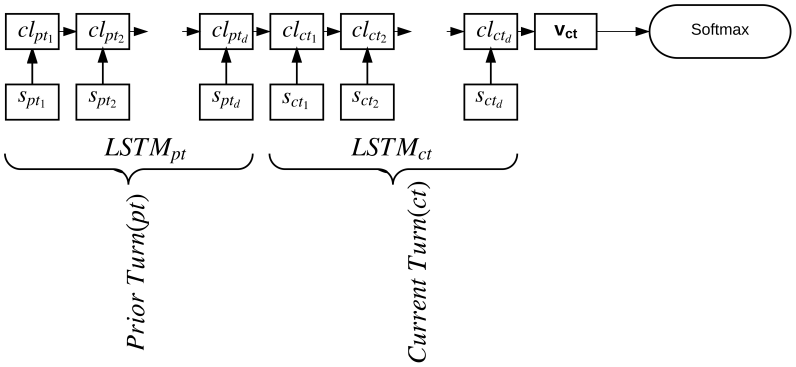


Figure 2
Conditional LSTM network for prior turn *pt* and current turn *ct*; figure is inspired by the model proposed in Rocktäschel et al. (2016).

Likewise, we compute v_{ct} for the current turn *ct* via $LSTM_{ct}$ (similar to Equation (2); also shown in Figure 1). Finally, we concatenate the vector v_{pt} and v_{ct} from the two LSTMs for the final softmax decision (i.e., predicting the *S* or *NS* class). In the event of using the succeeding turn *st* also in the model, we concatenate the vectors v_{pt}, v_{ct}, and v_{st}.

As stated earlier in this section, 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 5, however, we achieve the best performance using just the sentence-level attention. A possible explanation is that attention over both words and sentences seeks to learn a large number of model parameters and, given the moderate size of the discussion forum corpora, 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.

Conditional LSTM Networks. We also experiment with the *conditional encoding* model as introduced by Rocktäschel et al. (2016) for the task of recognizing textual entailment. In this architecture, two separate LSTMs are used—$LSTM_{pt}$ and $LSTM_{ct}$—similar to the previous architecture without any attention, but for $LSTM_{ct}$, its memory state is initialized with the last cell state of $LSTM_{pt}$. In other words, $LSTM_{ct}$ is conditioned on the representation of the $LSTM_{pt}$ that is built on the prior turn *pt*. For models that use the successive turn *st* as the context the LSTM representation $LSTM_{st}$ is conditioned on the representation of the $LSTM_{ct}$. Figure 2 shows the LSTM network where the current turn *ct* is conditioned on the prior turn *pt*.

Parameters and Pre-trained Word Vectors. All data sets were split randomly into training (80%), development (10%), and test (10%), maintaining the same distribution of sarcastic vs. non-sarcastic classes. For Twitter, we used the skip-gram word-embeddings (100-dimension) used in Ghosh, Guo, and Muresan (2015), which was built using over 2.5 million tweets.[8] For discussion forums, we use the standard Google *n*-gram *word2vec*

8 https://github.com/debanjanghosh/sarcasm_wsd.

pre-trained model (300-dimension) (Mikolov et al. 2013). Out-of-vocabulary words in the training set are randomly initialized via sampling values uniformly from (−0.05, 0.05) and optimized during training. We use the development data to tune the parameters (e.g., dropout rate, batch-size, number of epochs, *L*2-regularization) and selected dropout rate (Srivastava et al. 2014) of 0.5 (from [.25, 0.5, 0.75]), mini-batch size of 16, *L*2-regularization to 1E-4 (from [1E-2, 1E-3, 1E-4, 1E-5]), and set the number of epochs to 30. We set the threshold of the maximum number of sentences to ten per post so that for any post that is longer than ten sentences we select the first ten sentences for our experiments. Finally, the maximum number of words per sentence is set at 50 (zero-padding is used when necessary).

5. Results

In this section, we present a quantitative analysis aimed at addressing our first question, "does modeling conversation context help in sarcasm detection?" First, we consider just the *prior turn as conversation context* and show results of our various models on all data sets: IAC_{v2}, *Reddit*, and *Twitter* (Section 5.1). Also, we perform an experiment where we train on *Reddit* (discussion forum, self-labeled) and test on IAC_{v2} (discussion forum, labeled via crowdsourcing). Second, we consider *both the prior turn and the succeeding turn as context* and report results of various models on our IAC^{+}_{v2} data set (Section 5.2). We report Precision (P), Recall (R), and F1 scores on sarcastic (*S*) and non-sarcastic (*NS*) classes. We conclude the section with an error analysis of our models (Section 5.3).

5.1 Prior Turn as Conversation Context

We use two baselines, depicted as *bl*. First, disc^{ct}_{bl} and disc^{ct+pt}_{bl} represent the performance of the SVM models with discrete features (Section 4.1) when using only the current turn *ct* and the *ct* together with the prior turn *pt*, respectively. Second is the tf-idf based baseline. Here, tf-idf^{ct}_{bl} and $\text{tf-idf}^{ct+pt}_{bl}$ represent the performance of tf-idf values of *ct* and the *ct* together with the prior turn *pt*, respectively.

We experimented with both linear and RBF kernels and observed that the linear kernel consistently performed better than the RBF kernel. Only in the case of IAC_{v2}, for $\text{tf-idf}^{ct+pt}_{bl}$ setting, did the RBK kernel perform better (68.15% F1 for category *S* and 64.56% F1 for category *NS*). Thus, we only report the performance of the linear kernel for all the experiments.

Although we did not apply any feature selection, we use frequency threshold to select the *n*-grams (the minimum count is 5). Likewise, for the tf-idf based representation, the minimum frequency (i.e., DF) is set to 5. We use the development data to empirically select this minimum frequency. We also use the standard stop word list provided by the NLTK toolkit.

LSTM^{ct} and LSTM^{ct+pt} represent the performance of the simple LSTM models when using only the current turn *ct* and the *ct* concatenated with the prior turn *pt*, respectively. LSTM^{ct} + LSTM^{pt} depicts the multiple-LSTM model where one LSTM is applied on the current turn, *ct*, and the other LSTM is applied on the prior turn, *pt*. LSTM^{pt_a} and LSTM^{ct_a} are the attention-based LSTM models of context *pt* and current turn *ct*, where the w, s, and $w + s$ subscripts denote the word-level, sentence-level, or word- and sentence–level attentions. $\text{LSTM}^{conditional}$ is the *conditional encoding* model

that conditions the LSTM that reads the current turn on the LSTM that reads the prior turn (no attention). Given these notations, we present the results on each of the three data sets.

(IAC_{v2}) *Corpus.* Table 6 shows the classification results on the IAC_{v2} data set. Although a vast majority of the prior turn posts contain three to four sentences, around 100 have more than ten sentences and thus we set a cut-off to a maximum of ten sentences for context modeling. For the current turn *ct*, we consider the entire post.

We observe that neither of the baseline models, $disc_{bl}$ (based on discrete features) and $tf-idf_{bl}$ (based on tf-idf values) performed very well, and adding the context of the prior turn *pt* actually hurt the performance. Regarding the performance of the neural network models, we observed that the multiple-LSTMs model (one LSTM reads the context [*pt*] and one reads the current turn [*ct*], LSTM^{ct} + LSTM^{pt}) outperforms the model using just the current turn (results are statistically significant when compared with LSTM^{ct}). On the other hand, using only one LSTM to model both prior turn and current turn (LSTM^{ct+pt}) does not improve over just using the current turn, and has lower performance than the multiple-LSTM model (the results apply to the attention models as well). The highest performance when considering both the *S* and *NS* classes is achieved by the $\text{LSTM}^{conditional}$ model (73.32% F1 for *S* class and 70.56% F1 for *NS*, showing a 6 and 3 percentage point improvement over LSTM^{ct} for *S* and *NS* classes, respectively). The LSTM model with sentence-level attentions on both context and current turn ($\text{LSTM}^{ct_{a_s}}$+$\text{LSTM}^{pt_{a_s}}$) gives the best F1 score of 73.7% for the *S* class. For the *NS* class, although we notice an improvement in precision we also notice a drop in recall when compared with the LSTM model with sentence-level attention only on the current post ($\text{LSTM}^{ct_{a_s}}$). 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, attention is based on words and sentences, and we follow the architecture of hierarchical attention network (Yang et al. 2016). We observe that the performance (69.88% F1 for *S* category) deteriorates,

Table 6
Experimental results for the discussion forum data set (IAC_{v2}) (**bold** are best scores).

	S			*NS*		
Experiment	P	R	F1	P	R	F1
disc_{bl}^{ct}	65.55	66.67	66.10	66.10	64.96	65.52
dics_{bl}^{ct+pt}	63.32	61.97	62.63	62.77	64.10	63.50
tf-idf_{bl}^{ct}	66.07	63.25	64.63	64.75	67.52	66.11
$\text{tf-idf}_{bl}^{ct+pt}$	63.95	63.68	63.81	63.83	64.10	63.97
LSTM^{ct}	67.90	66.23	67.10	67.08	**68.80**	67.93
LSTM^{ct+pt}	65.16	67.95	66.53	66.52	63.68	65.07
LSTM^{ct}+LSTM^{pt}	66.19	79.49	72.23	74.33	59.40	66.03
$\text{LSTM}^{conditional}$	**70.03**	76.92	**73.32**	74.41	67.10	**70.56**
$\text{LSTM}^{ct_{a_s}}$	69.45	70.94	70.19	70.30	68.80	69.45
$\text{LSTM}^{ct_{a_s}+pt_{a_s}}$	64.46	69.33	66.81	67.61	62.61	65.01
$\text{LSTM}^{ct_{a_s}}$+$\text{LSTM}^{pt_{a_s}}$	66.90	**82.05**	**73.70**	**76.80**	59.40	66.99
$\text{LSTM}^{ct_{a_{w+s}}}$+$\text{LSTM}^{pt_{a_{w+s}}}$	65.90	74.35	69.88	70.59	61.53	65.75

probably because of the lack of enough training data. Since attention over both the words and sentences seeks to learn more model parameters, adding more training data will be helpful. For the *Reddit* and *Twitter* data (see subsequent sections), these models become better, but still not on par with just sentence-level attention, showing that even larger data sets might be needed.

Twitter Corpus. Table 7 shows the results of the Twitter data set. As with the IAC_{v2} data set, adding context using the discrete as well as the tf-idf features do not show a statistically significant improvement. For the neural networks models, similar to the results on the IAC_{v2} data set, the LSTM models that read both the context and the current turn outperform the LSTM model that reads only the current turn (LSTM^{ct}). However, unlike the IAC_{v2} corpus, for Twitter, we observe that for the LSTM without attention, the single LSTM architecture (i.e., LSTM^{ct+pt}) performs better, that is, 72% F1 between the sarcastic and non-sarcastic category (average), which is around 4 percentage points better than the multiple LSTMs (i.e., LSTM^{ct}+LSTM^{pt}). Since tweets are short texts, often the prior or the current turns are only a couple of words, hence concatenating the prior turn and current turn would give more context to the LSTM model. However, for sentence-level attention models, multiple-LSTMs are still a better choice than using a single LSTM and concatenating the context and current turn. The best performing architectures are again the LSTM^{conditional} and LSTM with sentence-level attentions (LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$). The LSTM^{conditional} model shows an improvement of 11 percentage point F1 on the *S* class and 4–5 percentage point F1 on the *NS* class, compared with LSTM^{ct}. For the attention-based models, the improvement using context is smaller (∼2% F1). We kept the maximum length of prior tweets to the last five tweets in the conversation context, when available. We also considered an experiment with only the "last" tweet (i.e., LSTM$^{ct_{a_s}}$+LSTM$^{last_pt_{a_s}}$), that is, considering only the "local conversation context" (see Section 3). We observe that although the F1 for the non-sarcastic category is high (76%), for the sarcastic category it is low (e.g., 71.3%). This shows that considering a

Table 7
Experimental results for Twitter data set (**bold** are best scores).

Experiment	*S*			*NS*		
	P	R	F1	P	R	F1
disc$^{ct}_{bl}$	64.20	64.95	64.57	69.0	68.30	68.70
disc$^{ct+pt}_{bl}$	65.64	65.86	65.75	70.11	69.91	70.00
tf-idf$^{ct}_{bl}$	63.16	67.94	65.46	70.04	65.41	67.64
tf-idf$^{ct+pt}_{bl}$	65.54	72.86	69.01	73.75	66.57	69.98
LSTM^{ct}	73.25	58.72	65.19	61.47	75.44	67.74
LSTM^{ct+pt}	70.54	71.19	70.80	64.65	74.06	**74.35**
LSTM^{ct}+LSTM^{pt}	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$^{ct_{a_s}}$	76.00	73.18	74.56	70.52	73.52	71.90
LSTM$^{ct_{a_s}+pt_{a_s}}$	70.44	67.28	68.82	72.52	75.36	**73.91**
LSTM$^{ct_{a_s}}$+LSTM$^{pt_{a_s}}$	**77.25**	75.51	**76.36**	72.65	**74.52**	**73.57**
LSTM$^{ct_{a_s}}$+LSTM$^{last_pt_{a_s}}$	73.10	69.69	71.36	74.58	**77.62**	**76.07**
LSTM$^{ct_{a_w}}$+LSTM$^{pt_{a_w}}$	76.74	69.77	73.09	68.63	75.77	72.02
LSTM$^{ct_{a_{w+s}}}$+LSTM$^{pt_{a_{w+s}}}$	76.42	71.37	73.81	69.50	74.77	72.04

Table 8
Experimental results for *Reddit* data set (**bold** are best scores).

Experiment	*S*			*NS*		
	P	R	F1	P	R	F1
disc_{bl}^{ct}	72.54	72.92	72.73	72.77	72.4	72.56
disc_{bl}^{ct+pt}	66.3	67.52	66.90	66.91	65.68	66.29
tf-idf_{bl}^{ct}	72.76	70.08	71.39	71.14	73.76	72.43
$\text{tf-idf}_{bl}^{ct+pt}$	71.14	69.72	70.42	70.31	71.72	71.01
LSTM^{ct}	**81.29**	59.6	68.77	68.1	**86.28**	**76.12**
LSTM^{ct+pt}	73.35	75.76	74.54	74.94	72.48	73.69
LSTM^{ct}+LSTM^{pt}	74.46	73.72	74.09	73.98	74.72	74.35
$\text{LSTM}^{conditional}$	73.72	71.6	72.64	72.40	74.48	73.42
$\text{LSTM}^{ct_{a_s}}$	74.87	74.28	74.58	74.48	75.08	**74.78**
$\text{LSTM}^{ct_{a_s}+pt_{a_s}}$	77.24	69.83	73.35	72.66	79.58	**75.96**
$\text{LSTM}^{ct_{a_s}}$+$\text{LSTM}^{pt_{a_s}}$	73.11	**80.60**	**76.67**	**78.39**	70.36	74.16
$\text{LSTM}^{ct_{a_{w+s}}}$+$\text{LSTM}^{pt_{a_{w+s}}}$	74.50	74.68	**74.59**	74.62	74.44	74.52

larger conversation context of multiple prior turns rather than just the last prior turn could assist in achieving higher accuracy, particularly in Twitter where each turn/tweet is short.

Reddit Corpus. Table 8 shows the results of the experiments on Reddit data. There are two major differences between this corpus and the IAC_{v2} corpus. First, because the original release of the Reddit corpus (Khodak, Saunshi, and Vodrahalli 2018) is very large, we select a subcorpus that is much larger than the IAC_{v2} data containing 50K instances. In addition, we selected posts (both *pt* and *ct*) that consist of a maximum of seven sentences primarily to be comparable with the IAC_{v2} data.[9] Second, unlike the IAC_{v2} corpus, the sarcastic current turns *ct* are self-labeled, so it is unknown whether there are any similarities between the nature of the data in the two discussion forums.

We observe that the baseline models (e.g., discrete as well as tf-idf features) perform similarly to the other discussion forum corpus IAC_{v2}. The disc_{bl}^{ct+pt} model performs poorly compared with the disc_{bl}^{ct} model. Note that Khodak, Saunshi, and Vodrahalli (2018) evaluated the sarcastic utterances via BoW features and sentence embeddings and achieved accuracy in the mid 70% range. However, they selected sentences between 2 and 50 words in length for the classification, which is very different from our set-ups, where we use larger comments (up to seven sentences).

Similar to the IAC_{v2} corpus, we observed that the multiple-LSTM models (one LSTM reads the context [*pt*] and one reads the current turn [*ct*]) outperform the models using just the current turn (results are statistically significant both for simple LSTM and LSTM with attentions). Multiple-LSTM with sentence-level attention performs best. Using one LSTM to model both prior turn and current turn has lower performance than the multiple-LSTM models.

We also conducted experiments with word and sentence-level attentions (i.e., $\text{LSTM}^{ct_{a_{w+s}}}$+$\text{LSTM}^{pt_{a_{w+s}}}$). Even though we obtain slightly lower accuracy (i.e., 76.67%

9 IAC_{v2} contains prior and current turns that contain mostly seven or fewer sentences.

for the sarcastic category) in comparison with sentence-level attention models, the difference is not as high as for the other corpora, which we believe is due to the larger size of the training data.

Impact of the Size and Nature of the Corpus. Overall, whereas the results on the *Reddit* data set are slightly better than on the IAC_{v2}, given that the *Reddit* corpus is ten times larger, we believe that the self-labeled nature of the *Reddit* data set might make the problem harder. To verify this hypothesis, we conducted two separate experiments. First, we selected a subset of the *Reddit* corpus that is equivalent to the IAC_{v2} corpus size (i.e., 5,000 examples balanced between the sarcastic and the not-sarcastic categories). We use the best LSTM model (i.e., attention on prior and current turn), which achieves 69.17% and 71.54% F1 for the sarcastic (*S*) and the non-sarcastic (*NS*) class, respectively. These results are lower than the ones we obtained for the IAC_{v2} corpus using the same amount of training data and much lower than the performances reported in Table 8. Second, we conducted an experiment where we trained our best models (i.e., LSTM models with sentence-level attention) on the *Reddit* corpus and tested on the test portion of the IAC_{v2} corpus. The results, shown in Table 9, are much lower than when training using ten times less data from the IAC_{v2} corpus, particularly for the sarcastic class (more than a 10 percentage point F1 measure drop). Moreover, unlike all the experiments, adding context does not help the classifier, which seems to highlight a difference between the nature of the two data sets, including the gold annotations (self-labeled for *Reddit* vs. crowdsource labeled for IAC_{v2}) and most likely the topics covered by these discussion forums.

Impact of Unbalanced Data Sets. In previous experiments we used a balanced data scenario. However, in online conversations we are most likely faced with an unbalanced problem (the sarcastic class is more rare than the non-sarcastic class). We thus experimented with an unbalanced setting, where we have more instances of the non-sarcastic class (*NS*) than sarcastic class (*S*) (e.g., two, three, or four times more data). We observe that the performance drops for the *S* category in the unbalanced settings, as expected. Table 10 shows the results of the unbalanced setting; particularly, we show the setting where the *NS* category has four times more training instances than the *S* category. We used the *Reddit* data set because it had a larger number of examples. For this experiment we used the best model from the balanced data scenario, which was the LSTM with sentence-level attention. In general, we observe that the Recall of the *S* category is low and that impacts the F1 score. During the LSTM training, class weights (inversely proportional to the sample sizes for each class) are added to the loss function to handle the unbalanced data scenario. We observe that adding contextual information

Table 9
Experimental results for training on the *Reddit* data set and testing on IAC_{v2} using the best LSTM models (sentence-level attention).

	S			*NS*		
Experiment	P	R	F1	P	R	F1
$LSTM^{ct_{a_s}}$	66.51	61.11	63.69	64.03	69.23	66.53
$LSTM^{ct_{a_s}}$ +$LSTM^{pt_{a_s}}$	63.96	60.68	62.28	62.60	65.81	64.17

Table 10
Experimental results for the *Reddit* data set under the unbalanced setting.

	S			*NS*		
Experiment	P	R	F1	P	R	F1
$\text{LSTM}^{ct_{a_s}}$	67.08	27.50	39.00	84.32	96.66	90.07
$\text{LSTM}^{ct_{a_s}}$ +$\text{LSTM}^{pt_{a_s}}$	62.25	35.05	44.85	85.48	94.73	89.87

(i.e., $\text{LSTM}^{ct_{a_s}}$ +$\text{LSTM}^{pt_{a_s}}$) helps the LSTM model and that pushes the F1 to 45% (i.e., a 6 point improvement over $\text{LSTM}^{ct_{a_s}}$).

5.2 Prior Turn and Subsequent Turn as Conversation Context

We also experiment using both the prior turn *pt* and the succeeding turn *st* as conversation context. Table 11 shows the experiments on the IAC^{+}_{v2} corpus. We observe that the performance of the LSTM models is high in general (i.e., F1 scores in between 78% and 84%, consistently for both the sarcastic [*S*] and non-sarcastic [*NS*] classes) compared with the discrete feature-based models (i.e., disc_{bl}). Table 11 shows that when we use conversation context, particularly the prior turn *pt* or the prior turn and the succeeding turn together, the performance improves (i.e., around 3 percentage point F1 improvement for sarcastic category and almost 6 percentage point F1 improvement for non-sarcastic category). For the *S* category, the highest F1 is achieved by the LSTM^{ct}+LSTM^{pt} model (i.e., 83.92%), whereas the LSTM^{ct}+LSTM^{pt}+LSTM^{st} model performs best for the non-sarcastic class (83.09%). Here, in the case of concatenating the turns and using a single LSTM (i.e., $\text{LSTM}^{ct+pt+st}$), the average F1 between the sarcastic and non-sarcastic category is 80.8%, which is around 3.5 percentage points lower than using separate LSTMs for separate turns (LSTM^{ct}+LSTM^{pt}+LSTM^{st}). In comparison to the attention-based models, although using attention over prior turn *pt* and successive turn *st* helps in sarcasm identification compared to the attention over only the current turn *ct* (i.e., improvement of around 2 percentage point F1 for both the sarcastic as well as the non-sarcastic class), generally the accuracy is slightly lower than the models without attention. We suspect this is because of the small size of the IAC^{+}_{v2} corpus ($<$ 3,000 instances).

We also observe that the numbers obtained for IAC^{+}_{v2} are higher than for the IAC_{v2} corpus even if less training data is used. To understand the difference, we analyzed the type of sarcastic and non-sarcastic posts from the IAC^{+}_{v2} and found that almost 94% of the corpus consists of sarcastic messages of "general" type, 5% of "rhetorical questions (*RQ*)" type and very few (0.6%) examples of the "hyperbolic" type (Oraby et al. 2016). Looking at Oraby et al. (2016), it seems that the "general" type obtains the best results (Table 7 in Oraby et al. [2016]), with almost 10 percentage point F1 over the "hyperbolic" type. As we stated before, although the IAC_{v2} corpus is larger than the IAC^{+}_{v2} corpus, IAC_{v2} maintains exactly the same distribution of "general," "*RQ*," and "hyperbolic" examples. This also explains why Table 11 shows superior results, since classifying the "generic" type of sarcasm could be an easier task.

Table 11
Experimental results for the $IAC_{v2_{st}}$ data set using prior and succeeding turns as context (**bold** are best scores).

Experiment	*S*			*NS*		
	P	R	F1	P	R	F1
disc_{bl}^{ct}	76.97	78.67	77.81	78.83	77.14	77.97
disc_{bl}^{ct+pt}	76.69	75.0	75.83	76.22	77.85	77.03
disc_{bl}^{ct+st}	67.36	71.32	69.28	70.45	66.43	68.38
$\text{disc}_{bl}^{ct+pt+st}$	74.02	69.12	71.48	71.81	76.43	74.05
tf-idf_{bl}^{ct}	71.97	69.85	70.90	71.53	73.57	72.54
$\text{tf-idf}_{bl}^{ct+pt}$	72.66	74.26	73.45	74.45	72.86	73.65
$\text{tf-idf}_{bl}^{ct+st}$	72.73	70.59	71.64	72.22	74.29	73.24
$\text{tf-idf}_{bl}^{ct+pt+st}$	75.97	72.06	73.96	74.15	77.86	75.96
LSTM^{ct}	74.84	87.50	80.68	85.47	71.43	77.82
LSTM^{ct+pt}	69.03	78.67	73.53	76.03	65.71	70.49
LSTM^{ct+st}	78.38	85.29	81.60	84.37	77.14	80.59
$\text{LSTM}^{ct+pt+st}$	76.62	88.06	81.94	86.55	74.10	79.84
$\text{LSTM}^{ct}+\text{LSTM}^{pt}$	80.00	**88.24**	**83.92**	**87.30**	78.57	**82.71**
$\text{LSTM}^{ct}+\text{LSTM}^{st}$	79.73	86.76	83.10	85.94	78.57	82.09
$\text{LSTM}^{ct}+\text{LSTM}^{pt}+\text{LSTM}^{st}$	**81.25**	86.03	**83.57**	85.61	80.71	**83.09**
$\text{LSTM}^{conditional(pt->ct)}$	79.26	78.68	78.97	79.43	80.00	79.71
$\text{LSTM}^{conditional(ct->st)}$	70.89	69.85	70.37	71.13	72.14	71.63
$\text{LSTM}^{ct_{a_s}}$	77.18	84.56	80.70	83.46	75.71	79.40
$\text{LSTM}^{ct_{a_s}}+\text{LSTM}^{pt_{a_s}}$	**80.14**	83.09	81.59	82.96	**80.00**	81.45
$\text{LSTM}^{ct_{a_s}}+\text{LSTM}^{st_{a_s}}$	75.78	**89.71**	82.15	**87.83**	72.14	79.22
$\text{LSTM}^{ct_{a_s}}+\text{LSTM}^{pt_{a_s}}+\text{LSTM}^{st_{a_s}}$	76.58	**88.97**	82.31	87.29	73.57	79.84
$\text{LSTM}^{ct_{a_{w+s}}}+\text{LSTM}^{pt_{a_{w+s}}}$	79.00	80.14	79.56	80.43	79.29	79.86

5.3 Error Analysis

We conducted an error analysis of our models and identified the following types.

Missing Background Knowledge. Sarcasm or verbal irony depends to a large extent upon the shared knowledge of the speaker and hearer (common ground) that is not explicitly part of the conversation context (Haverkate 1990). For instance, notice the following context/sarcastic reply pair from the IAC_{v2} corpus.

> **userA:** i'm not disguising one thing. I am always clear that my argument is equal marriage for same sex couples. No one i know on my side argues simply for "equality in marriage".
>
> **userB:** Right, expect when talking about the 14th amendment, The way you guys like to define "equal protection" would make it so any restriction is unequal.

Here, **userB** is sarcastic while discussing the 14th amendment (i.e., equal protection). On social media, users often argue about different topics, including controversial ones.[10] When engaged in conversations, speakers might assume that some background

10 As stated earlier, *IAC* includes a large set of conversations from 4forums.com, a Web site for political debates (Walker et al. 2012; Justo et al. 2014).

knowledge about those topics is understood by the hearers (e.g., historical events, constitution, politics). For example, posts from *Reddit* are based on specific subreddits where users share similar interests (i.e., video games). We found, often, that even if the sarcastic posts are not political, they are based on specific shared knowledge (e.g., the performance of a soccer player in recent games). LSTM or SVM models are unable to identify the sarcastic intent when such contextual knowledge that is outside of the conversation context is used by the speaker. In the future, however, we intend to build a model on specific subreddits (i.e., politics, sports) to investigate how much the domain-specific knowledge helps the classifiers.

Longer Sarcastic Reply. Although, the IAC_{v2} and *Reddit* corpora are annotated differently (using crowdsourcing vs. self-labeled, respectively), the labels are for the posts and not for specific sentences. Thus for longer posts, often the LSTM models perform poorly because the sarcastic cue is buried under the remaining non-sarcastic parts of the post. For instance, we observe that about 75% of the false negative cases reported by the $\text{LSTM}^{c_{a_s}}$+$\text{LSTM}^{r_{a_s}}$ on the IAC_{v2} data have five or more sentences in the sarcastic posts.

Use of Profanity and Slang. Sarcasm could be bitter, caustic, snarky, or could have a mocking intent. Oraby et al. (2016) asked the annotators to look for such characteristics while annotating the IAC_{v2} posts for sarcasm. We observe that although the LSTM models are particularly efficient in identifying some inherent characteristics of sarcastic messages such as "context incongruity" (detailed in Section 6), they often miss the sarcastic posts that contain slang and the use of profane words. In the future, we plan to utilize a lexicon similar to Burfoot and Baldwin (2009) to identify such posts.

Use of Numbers. In some instances, sarcasm is related to situations that involve numbers, and the models are unable to identify such cases (i.e., userB: "why not? My mother has been 39, for the last 39 years." in reply of userA: "actually the earth is 150 years old. fact and its age never changes"). This type of sarcasm often occurs in social media both in discussion forums and on Twitter (Joshi, Sharma, and Bhattacharyya 2015).

Use of Rhetorical Questions. We also found that sarcastic utterances that use rhetorical questions (*RQ*), especially in discussion forums (e.g., IAC_{v2}) are hard to identify. Oraby et al. (2016) hypothesized that sarcastic utterances of *RQ* type are of the following structure: They contain questions in the middle of a post that are followed by a statement. Because many discussion posts are long and might include multiple questions, question marks are not very strong indicators for *RQ*.

6. Qualitative Analysis

Wallace et al. (2014) showed that by providing additional conversation context, humans could identify sarcastic utterances that they were unable to identify without the context. However, it will be useful to understand whether a specific *part of the conversation context triggers* the sarcastic reply. To begin to address this issue, we conducted a qualitative study to understand (a) whether human annotators can identify parts of context that trigger the sarcastic reply and (b) if attention weights can signal similar information. For (a) we designed a crowdsourcing experiment (Crowdsourcing Experiment 1 in Section 6.1), and for (b) we looked at the attention weights of the LSTM networks (Section 6.2).

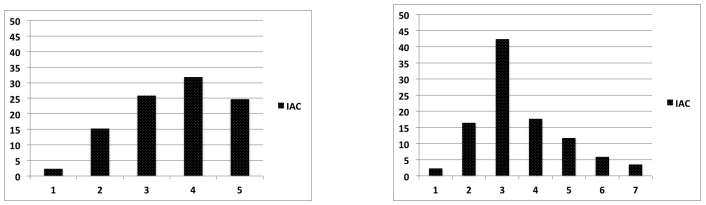
In addition, discussion forum posts are usually long (several sentences), and we noticed in our error analysis that computational models have a harder time in correctly labeling these as sarcastic or not. The second issue we want to investigate is whether there is a particular sentence in the sarcastic post that expresses the speaker's sarcastic intent. To begin to address this issue, we conducted another qualitative study to understand (a) whether human annotators can identify a sentence in the sarcastic post that mainly expresses the speaker's sarcastic intent and (b) if the sentence-level attention weights can signal similar information. For (a) we designed a crowdsourcing experiment (Crowdsourcing Experiment 2 in Section 6.1), and for (b) we looked at the attention weights of the LSTM networks (Section 6.2).

For both studies, we compare the human annotators' selections with the attention weights to examine whether the attention weights of the LSTM networks are correlated to human annotations.

6.1 Crowdsourcing Experiments

Crowdsourcing Experiment 1. We designed an Amazon Mechanical Turk task (for brevity, MTurk) as follows: given a pair of a sarcastic current turn (C_TURN) and its prior turn (P_TURN), we ask Turkers to identify one or more sentences in P_TURN that they think triggered the sarcastic reply. Turkers could select one or more sentences from the conversation context P_TURN, including the entire turn. We selected all sarcastic examples from the IAC_{v2} test set where the prior turn contain between three and seven sentences, because longer turns might be a more complex task for the Turkers. This selection resulted in 85 pairs. We provided several definitions of sarcasm to capture all characteristics. The first definition is inspired by the Standard Pragmatic Model (Grice, Cole, and Morgan 1975), which identifies verbal irony or sarcasm as a speech or form of writing that means the opposite of what it seems to say. In another definition, taken from Oraby et al. (2016), we mentioned that sarcasm often is used with the intention to mock or insult someone or to be funny. We provided a couple of examples of sarcasm from the IAC_{v2} data set to show how to successfully complete the task (See Appendix A for the instructions given the the Turkers). Each HIT contains only one pair of C_TURN and P_TURN and five Turkers were allowed to attempt each HIT. Turkers with reasonable quality (i.e., more than 95% of acceptance rate with experience of over 8,000 HITs) were selected and paid $0.07 per task. Because Turkers were asked to select one or multiple sentences from the prior turn, standard interannotator agreement (IAA) metrics are not applicable. Instead, we look at two aspects to understand the user annotations. First, we look at the distribution of the triggers (i.e., sentences that trigger the sarcastic reply) selected by the five annotators (Figure 3). It can be seen that in 3% of instances all five annotators selected the exact same trigger(s), while in 58% of instances 3 or 4 different selections were made per posts. Second, we looked at the distribution of the number of sentences in the P_TURN that were selected as triggers by Turkers. We notice that 43% of the time three sentences were selected.

Crowdsourcing Experiment 2. The second study is an extension of the first study. Given a pair of a sarcastic turn C_TURN and its prior turn P_TURN, we ask the Turkers to perform two subtasks. First, they were asked to identify "only one" sentence from C_TURN that expresses the speaker's sarcastic intent. Next, based on the selected sarcastic sentence, they were asked to identify one or more sentences in P_TURN that may trigger that sarcastic sentence (similar to the Crowdsourcing Experiment 1). We selected examples both from the IAC_{v2} corpus (60 pairs) as well as the *Reddit* corpus (100 pairs).



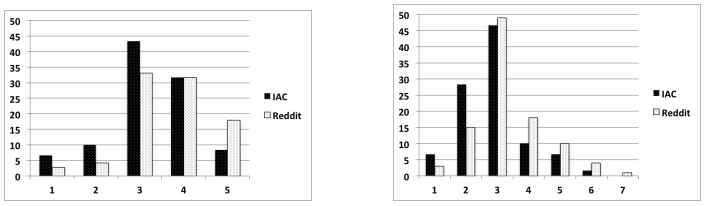
(a) (b)

Figure 3
Crowdsourcing Experiment 1: (a) number of different trigger selections made by the five Turkers (1 means all Turkers selected the exact same trigger(s)) and (b) distribution of the number of sentences chosen by the Turkers as triggers in a given post; both in %.

Each of the P_TURN and C_TURN contains three to seven sentences (note that the examples from the IAC_{v2} corpus are a subset of the ones used in the previous experiment). We replicate the same design as the previous MTurk (i.e., we included definitions of sarcasm, provided examples of the task, used only one pair of C_TURN and P_TURN per HIT, required the same qualification for the Turkers, and paid the same payment of $0.07 per HIT; see Appendix A for the instructions given to Turkers). Each HIT was done by five Turkers (a total of 160 HITs). To measure the IAA between the Turkers for the first subtask (i.e., identifying a particular sentence from C_TURN that expresses the speaker's sarcastic intent) we used Krippendorf's α (Krippendorff 2012). We measure IAA on nominal data (i.e., each sentence is treated as a separate category). Because the number of sentences (i.e., categories) can vary between three and seven, we report separate α scores based on the number of sentences. For C_TURN that contains three, four, five, or more than five sentences, the α scores are 0.66, 0.71, 0.65, 0.72, respectively. The α scores are modest and illustrate (a) identifying sarcastic sentences from a discussion forum post is a hard task and (b) it is plausible that the current turn (C_TURN) contains multiple sarcastic sentences. For the second subtask, we carried a similar analysis as for Experiment 1, and results are shown in Figure 4 both for the IAC_{v2} and *Reddit* data.

6.2 Comparing Turkers' Answers with the Attention Weight of the LSTM Models

In this section, we compare the Turkers' answers for both tasks with the sentence-level attention weights of the LSTM models. This analysis is an attempt to provide an interpretation of the attention mechanism of the LSTM models for this task.



(a) (b)

Figure 4
Crowdsourcing Experiment 2: (a) number of different trigger selections made by the five Turkers (1 means all Turkers selected the exact same trigger(s)) and (b) distribution of the number of sentences chosen by the Turkers as triggers in a given post; both in %.

S1	Ok...
S2	I have to stop to take issue with something here, that I see all too often.
S3	And I've held my tongue on this as long as I can

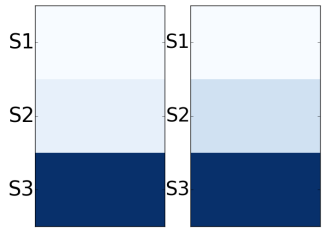


Figure 5
Sentences in P_TURN; heatmap of the *attention weights* (left) and *Turkers' selection* (right) of which of those sentences trigger the sarcastic C_TURN = "Well, it's not as though you hold your tongue all that often when it serves in support of an anti-gay argument."

To identify what part of the prior turn triggers the sarcastic reply, we first measure the overlap of Turkers' choice with the sentence-level attention weights of the $\text{LSTM}^{ct_{a_s}}$+$\text{LSTM}^{pt_{a_s}}$ model. For Crowdsourcing Experiment 1, we used the models that are trained/tested on the IAC_{v2} corpus. We selected the sentence with the highest attention weight and matched it to the sentence selected by Turkers using majority voting. We found that 41% of the time the sentence with the highest attention weight is also the one picked by Turkers. Figures 5 and 6 show side by side the heat maps of the attention weights of LSTM models (left hand side) and Turkers' choices when picking up sentences from the prior turn that they thought triggered the sarcastic reply (right hand side). For Crowdsourcing Experiment 2, 51% and 30% of the time the sentence with the highest attention weight is also the one picked by Turker for IAC_{v2} and *Reddit*, respectively.

To identify what sentence of the sarcastic current turn expresses best the speaker's sarcastic intent, we again measure the overlap of Turkers' choice with the sentence-level attention weights of the $\text{LSTM}^{ct_{a_s}}$+$\text{LSTM}^{pt_{a_s}}$ model (looking at the sentence-level attention weights from the current turn). We selected the sentence with the highest attention weight and matched it to the sentence selected by Turkers using majority voting. For IAC_{v2}, we found that 25% of the time the sentence with the highest attention weight is also the one picked by the Turkers. For *Reddit*, 13% of the time the sentence with the highest attention weight is also the one picked by the Turkers. The low agreement on *Reddit* illustrates that many posts may contain multiple sarcastic sentences.

S1	How do we rationally explain these creatures existence …for millions of years?
S2	and if they were the imaginings of bronze age …we can now recognize from fossil evidence?
S3	and while your at it …200 million years without becoming dust?

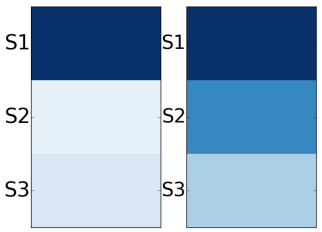


Figure 6
Sentences in P_TURN (userC in Table 1); heatmap of the *attention weights* (left) and *Turkers' selection* (right) of which of those sentences trigger the sarcastic C_TURN (userD in Table 1).

For both of these issues, 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 to answer this question.

6.3 Interpretation of the Turkers' Answers and the Attention Models

We visualize and compare the sentence-level as well as the word-level attention weights of the LSTM models with the Turkers' annotations.

Semantic Coherence Between Prior Turn and Current Turn. Figure 5 shows a case where the prior turn contains three sentences, and the sentence-level attention weights are similar to the Turkers' choice of what sentence(s) triggered the sarcastic turn. Looking at this example, it seems the model pays attention to output vectors that are *semantically coherent* between P_TURN and C_TURN. The sarcastic C_TURN of this example contains a single sentence—"Well, it's not as though you hold your tongue all that often when it serves in support of an anti-gay argument"—while the sentence from the prior turn P_TURN that received the highest attention weight is S3—"And I've held my tongue on this as long as I can."

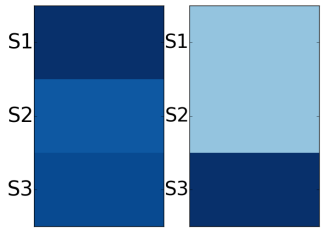
In Figure 6, the highest attention weight is given to the most informative sentence—"how do we rationally explain these creatures existence so recently in our human history if they were extinct for millions of years?" Here, the sarcastic post C_TURN (userD's post in Table 1) mocks userC's prior post ("how about this explanation – you're reading waaaaay too much into your precious bible"). For both Figure 5 and Figure 6, the sentence from the prior turn P_TURN that received the highest attention weight has also been selected by the majority of the Turkers. For Figure 5 the distribution of the attention weights and Turkers' selections are alike. Both examples are taken from the IAC_{v2} corpus.

Figure 7 shows a pair of conversation context (i.e., prior turn) and the sarcastic turn (userE's and userF's posts in Table 1), together with their respective heatmaps, which reflect the two subtasks performed in the second crowdsourcing experiment. The bottom part of the figure represents the sentences from the C_TURN and the heatmaps that compare attention weights and the Turkers' selections for the first subtask: selecting the sentence from C_TURN that best expresses the speaker's sarcastic intent. The top part of the figure shows the sentences from the P_TURN as well as the heatmaps to show what sentence(s) are more likely to trigger the sarcastic reply. We make two observations: (a) Different Turkers selected different sentences from the C_TURN as expressing sarcasm. The attention model has given the highest weight to the last sentence in C_TURN, similar to the Turkers's choice. (b) The attention weights seem to indicate semantic coherence between the sarcastic post (i.e., "nothing to see here" with the prior turn "nothing will happen, this is going to die . . . ").

We also observe similar behavior in tweets (highest attention to words –*majority* and *gerrymandering* later in Figure 9(d)).

*Incongruity Between Conversation Context (*P_TURN*) and Current Turn (*C_TURN*).* Context incongruity is an inherent characteristic of irony and sarcasm and has been extensively studied in linguistics, philosophy, communication science (Grice, Cole, and Morgan 1975; Attardo 2000; Burgers, Van Mulken, and Schellens 2012) as well as recently in NLP (Riloff et al. 2013; Joshi, Sharma, and Bhattacharyya 2015). It is possible that the literal meaning of the current turn C_TURN is incongruent with the conversation

P_TURN	S1	nothing will happen, . . . other private member motions.
	S2	this whole thing is being made . . . are trying to push an agenda.
	S3	feel free to let your . . . discussion before it is send to the trashcan.



C_TURN	S1	the usual "nothing to see here" response.
	S2	whew!
	S3	we can sleep at night and ignore this.

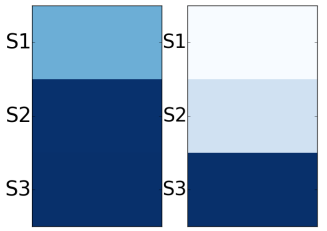


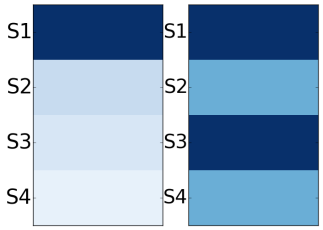
Figure 7
Sentences from P_TURN that trigger sarcasm (top) and sentences from C_TURN that express sarcasm (bottom). Tables show, respectively, the text from P_TURN and C_TURN (top and bottom) and figure shows the heatmap of *attention weights* (left) and *Turkers' selection* (right).

context (P_TURN). We observe in discussion forums and Twitter that the attention-based models have frequently identified sentences and words from P_TURN and C_TURN that are semantically incongruous. For instance, in Figure 8, the attention model has given more weight to sentence S2 ("protecting your home from a looter?") in the current turn, whereas from the P_TURN the model assigned the highest weight to sentence S1 ("'this guy chose to fight in the ukraine"). Here the model picked up the opposite sentiment from the P_TURN and C_TURN, that is, "chose to fight" and "protecting home from looter." Thus, the model seems to learn the incongruity between the prior turn P_TURN and the current turn C_TURN regarding the opposite sentiment. Also, the attention model selects (i.e., second highest weight) sentence S2 from the P_TURN ("he died because of it"), which also shows that the model captures opposite sentiment between the conversation context and the sarcastic post.

However, from Figure 8, we notice that some of the Turkers choose the third sentence S3 ("sure russia fuels the conflict, but he didnt have to go there") in addition to sentence S1 from the context P_TURN. Here, the Turkers utilize their background knowledge on global political conflicts (see Section 5.3) to understand the context incongruity, a fact missed by the attention model.

In the Twitter data set, we observe that the attention models often have selected utterance(s) from the context that have the opposite sentiment (Figure 9(a), Figure 9(b), and Figure 9(c)). 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" vs. "gutsy"). Word-models seem to also work well when words in the prior turn and current turn are semantically incongruous but not related to sentiment ("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 9(d)).

P_TURN	S1	technically speaking : this guy chose to fight in the ukraine.
	S2	he died because of it.
	S3	sure russia fuels the conflict, but he have to go there.
	S4	his choice, his consequences.



C_TURN	S1	sure thing.
	S2	protecting your home from an looter?
	S3	nope, why would anyone do that?

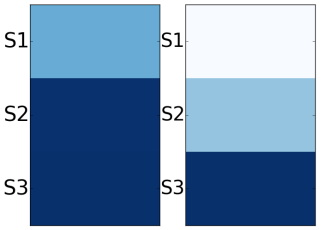


Figure 8
Sentences from P_TURN that trigger sarcasm (top) and sentences from C_TURN that represents sarcasm (bottom). Tables show respectively the text from P_TURN and C_TURN (top and bottom) and figure shows *attention weights* (LHS) and *Turkers' selection* (RHS).

Attention Weights and Sarcasm Markers. Looking just at the attention weights in the replies, we notice that the models are giving the highest weight to sentences that contain sarcasm markers, such as emoticons (e.g., ":p", ":)") and interjections (e.g., "ah", "hmm"). We also observe that interjections such as "whew" with exclamation mark receive high attention weights (Figure 7; see the attention heatmap for the current turn C_TURN). Sarcasm markers such as the use of emoticons, uppercase spelling of words, or interjections, are explicit indicators of sarcasm that signal that an utterance is sarcastic (Attardo 2000; Burgers, Van Mulken, and Schellens 2012; Ghosh and Muresan 2018). Use of such markers in social media (mainly on Twitter) is extensive.

Reversal of Valence. The reversal of valence is an essential criterion of sarcastic messages that states that the intended meaning of the sarcastic statement is opposite to its literal meaning (Burgers 2010). One of the common ways of representing sarcasm is through sarcastic praise (i.e., sarcasm with a positive literal meaning as in "Great game, Bob!", when the game was poor) and sarcastic blame (i.e., sarcasm with a negative literal meaning as in "Horrible game, Bob!", when the game was great). Ghosh, Guo, and Muresan (2015) have studied the use of words that are used extensively in social media, particularly on Twitter to represent sarcastic praise and blame. For instance, words such as "genius" and "best" are common in representing sarcastic praise because we need to alter their literal to intended meaning to identify the sarcasm. In our analysis, we often observe that the attention models have put the highest weights to such terms (i.e., "greatest," "mature," "perfect") whose intended use in the sarcastic statement is opposite to its literal meaning.

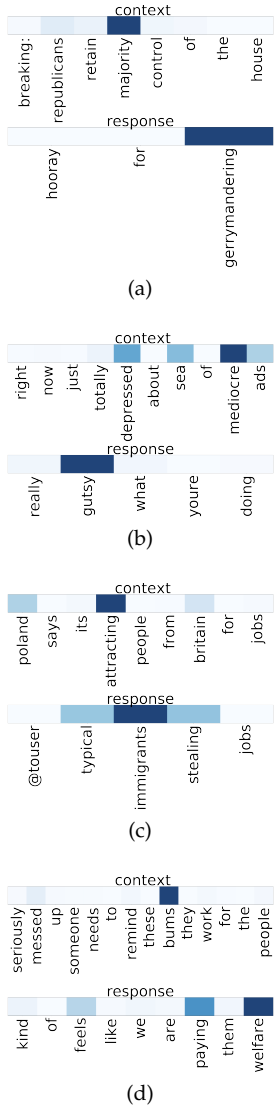


Figure 9
Attention visualization of incongruity between P_TURNs and C_TURNs on *Twitter*.

7. Conclusions and Future Directions

This research makes a complementary contribution to existing work on modeling context for sarcasm/irony detection by looking at a particular type of context, *conversation context*. We have modeled both the prior and succeeding turns when available as conversation context. Although Twitter is the de facto platform for research on verbal irony or sarcasm, we have thoroughly analyzed both Twitter and discussion forum data sets.

We have addressed three questions:

1. *Does modeling of conversation context help in sarcasm detection?* To answer this question, we show that only if we explicitly model the context and the current turn using a multiple-LSTM architecture do we obtain

improved results as compared with just modeling the current turn. The multiple-LSTM architecture is designed to recognize a possible inherent incongruity between the current turn and the context, and thus it is important to keep the C_TURN and the context (P_TURN and/or S_TURN) separate as long as possible. This incongruity might become diffuse if the inputs are combined too soon, and we have shown that the multiple-LSTM architecture outperforms a single LSTM architecture that combines the current turn and the context. In particular, LSTM networks with sentence-level attention achieved significant improvement when using prior turn as context for all the data sets (e.g., 6–11 percentage point F1 for IAC_{v2} and Twitter messages). Using the succeeding turn did not prove to be helpful for our data sets.

2. *Can humans and computational models determine what part of the conversation context (P_TURN) triggered the sarcastic reply (C_TURN)?* To answer this question, we conducted a qualitative study to understand (a) whether human annotators can identify parts of the context that trigger the sarcastic reply and (b) if the attention weights of the LSTM models can signal similar information. This study also constitutes an attempt to provide an interpretation of the attention mechanism of the LSTM models for our task. Our results show, in Crowdsourcing Experiment 1, that for 41% of the time the sentence with the highest attention weight is also the one picked by the Turkers.

3. *Given a sarcastic post that contains multiple sentences, is it feasible to identify a particular sentence that expresses the speaker's sarcastic intent?* To answer this question we conducted another qualitative study to understand (a) whether human annotators can identify a sentence in the sarcastic post that mainly expresses the speaker's sarcastic intent and (b) if the sentence-level attention weights can signal similar information. This study again aimed to provide an interpretation of the attention mechanism of the LSTM models. For this task, the agreement between the attention weights of the models and humans (using majority voting) is lower than for the previous task. However, the IAA between Turkers is also just moderate (α between 0.66 and 0.72), which shows that this is inherently a difficult task. It might also be the case that a post/turn is sarcastic in general and not a single sentence can be selected as being the only sarcastic piece.

Our experiments showed that attention-based models can identify inherent characteristics of sarcasm (i.e., sarcasm markers and sarcasm factors such as context incongruity). We also conducted a thorough error analysis and identified several types of errors: missing world knowledge, use of slang, use of rhetorical questions, and use of numbers. In future work, we plan to develop approaches to tackle these errors, such as modeling rhetorical questions (similar to Oraby et al. [2017]), having a specialized approach to model sarcastic messages related to numbers, or using additional lexicon-based features to include slang.

Although a few groups have conducted recent experiments on discussion forum data, we understand that there are many questions to address here. First, we show that self-labeled sarcastic turns (e.g., *Reddit*) are harder to identify compared with a corpus where turns are externally annotated (crowdsourced) (e.g., IAC_{v2}). We show that even

if the training data in *Reddit* is ten times larger, it did not make much impact in our experiments. However, the *Reddit* corpus consists of several subreddits, so it might be interesting in the future to experiment with training data from a particular genre of subreddit (e.g., political forums). Second, during crowdsourcing, the Turkers are provided with the definition(s) of the phenomenon under study, which is not applicable in self-labeled corpora. It is unclear whether authors of sarcastic or ironic posts are using any specific definition of sarcasm or irony while labeling (and we see ironic posts labeled with the #sarcasm hashtag).

In future work we plan to study the impact of using a larger context such as the full thread in a discussion, similar to Zayats and Ostendorf (2018). This will also be useful in order to gain a broader understanding of the role of sarcasm in social media discussions (i.e., sarcasm as a persuasive strategy). We are also interested in utilizing external background knowledge to model sentiment about common situations (e.g., going to the doctor; being alone) or events (e.g., rainy weather) that users are often sarcastic about.

Appendix A. Mechanical Turk Instructions

A.1 Crowdsourcing Experiment 1

Identify what triggers a sarcastic reply. Sarcasm is a speech or form of writing that means the opposite of what it seems to say. Sarcasm is usually intended to mock or insult someone or to be funny. People participating in social media platforms, such as discussion forums, are often sarcastic. In this experiment, a pair of posts (previous post and sarcastic reply) from an online discussion forum is presented to you. The sarcastic reply is a response to the previous post. However, given that these posts may contain more than one sentence, often sarcasm in the sarcastic reply is triggered by only one or just a few of the sentences from the previous post.

Your task will be to identify the sentence/sentences from the previous post that triggers the sarcasm in the sarcastic reply. Consider the following pair of posts (sentence numbers are in "()").

- ***UserA: previous post***: (1) It's not just in case of an emergency. (2) It's for everyday life. (3) When I have to learn Spanish just to order a burger at the local Micky Dee's, that's a problem. (4) Should an English speaker learn to speak Spanish if they're going to Miami?
- ***UserB: sarcastic reply***: When do you advocate breeding blond haired, blue eyed citizens to purify the US?

Here, UserB's sarcastic reply is triggered by sentence 3 ("When I have to learn Spanish…") and sentence 4 ("Should an English speaker…") from UserA's post and not the other sentences in the post.

DESCRIPTION OF THE TASK. Given such a pair of online posts, your task is to identify the sentences from the previous post that trigger sarcasm in the sarcastic reply. You only need to select the sentence numbers from the previous post (do not retype the sentences).

EXAMPLES. Here are some examples of how to perform the task.

Example 1

- ***UserA: previous post***: (1) see for yourselves. (2) The fact remains that in the caribbean, poverty and crime was near nil. (3) Everyone was self-sufficient and contented with the standard of life. (4) there were no huge social gaps.
- ***UserB: sarcastic reply***: Are you kidding me?! You think that Caribbean countries are "content?!" Maybe you should wander off the beach sometime and see for yourself.
- **Answers**: 2,3.

Example 2

- ***UserA: previous post***: (1) Sure I can! (2) That is easy. (3) Bible has lasted thousands of years under the unending scrutiny of being judged by every historical discovery. (4) Never has it been shown to be fictional or false.
- ***UserB: sarcastic reply***: Except for, ya know, like the whole Old Testament ;) False testament: archaeology refutes the Bible's claim to history.
- **Answers**: 3,4.

A.2 Crowdsourcing Experiment 2

Identify what triggers a sarcastic reply. Sarcasm is a speech or form of writing that means the opposite of what it seems to say. Sarcasm is usually intended to mock or insult someone or to be funny. People participating in social media platforms, such as discussion forums, are often sarcastic.

In this experiment, a pair of posts (previous post and sarcastic post) from an online discussion forum is presented to you. Suppose the authors of the posts are, respectively, UserA and UserB. The sarcastic post from UserB is a response to the previous post from UserA. Your task is twofold. First, from UserB's sarcastic post you have to identify the particular "sentence" that presents sarcasm. Remember, you need to select only ONE sentence here. Next, given this sarcastic sentence look back at UserA's post. Often sarcasm in the sarcastic reply is triggered by only one or just a few of the sentences from the previous post. Your second task is to identify the sentence/sentences from the UserA's post that triggers the sarcasm in UserB's post.

Consider the following pair of posts (sentence numbers are in "()").

- ***UserA: previous post***: (1) see for yourselves. (2) The fact remains that in the caribbean, poverty and crime was near nil. (3) Everyone was self-sufficient and contented with the standard of life. (4) there were no huge social gaps.
- ***UserB: sarcastic reply***: (1) Are you kidding me? (2) You think that Caribbean countries are "content?" (3) Maybe you should wander off the beach sometime and see for yourself. Here, the sarcastic sentence in the sarcastic post of UserB is the third sentence ("maybe you should wander off the beach. . . ")

At the same time, UserB is sarcastic on the previous post from UserA and the sarcasm is triggered by sentence 2 ("Caribbean, poverty and crime was near nil . . . ") and sentence 3 ("and everyone was self-sufficient . . . ") and not the other sentences in the post.

DESCRIPTION OF THE TASK. Given such a pair of online posts, your task is twofold. First, you need to identify the sentence (i.e., only one sentence) from UserB's sarcastic reply that presents sarcasm. Next, from UserA's post select the sentences that trigger sarcasm in UserB's post. For both tasks you only need to select the sentence number (do not retype the sentences).

EXAMPLES. Here are some examples of how to perform the task.

Example 1

- ***UserA: previous post***: (1) Sure I can! (2) That is easy. (3) Bible has lasted thousands of years under the unending scrutiny of being judged by every historical discovery. (4) Never has it been shown to be fictional or false.
- ***UserB: sarcastic reply***: (1) Except for, ya know, like the whole Old Testament ;) (2) False testament: archaeology refutes the Bible's claim to history.
- **Answers** (Sarcastic Sentence in UserB sarcastic reply): 1.
- **Answers** (Sentences from UserA's post that trigger the sarcastic sentence in userB's reply): 3, 4.

Example 2

- ***UserA: previous post***: (1) hasn't everyday since christ been latter days, thousands of days and he hasn't returned as promised. (2) in the bible his return was right around the corner ... how many years has it been. (3) when will you realize he isn't coming back for you!
- ***UserB: sarcastic reply***: (1) how about when it dawns on you who he was when he came the first time? (2) lol (3) trade in your blinders for some spiritual light!
- **Answers** (Sarcastic Sentence in UserB sarcastic reply): 3
- **Answers** (Sentences from UserA's post that trigger the sarcastic sentence in userB's reply): 1, 2, 3.

Acknowledgments

This article 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

Amir, Silvio, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mario J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin.

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

Bamman, David and Noah A. Smith. 2015. Contextualized sarcasm detection on Twitter. In *Ninth International AAAI Conference on Web and Social Media*, pages 574–577, Oxford.

Bharti, Santosh Kumar, Korra Sathya Babu, and Sanjay Kumar Jena. 2017. Harnessing online news for sarcasm detection in Hindi tweets. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 679–686.

Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Bowman, Samuel R., Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon.

Burfoot, Clint and Timothy Baldwin. 2009. Automatic satire detection: Are you having a laugh? In *Proceedings of the ACL-IJCNLP 2009 Conference*, pages 161–164, Singapore.

Burgers, Christian, 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.

Burgers, Christian Frederik. 2010. *Verbal Irony: Use and Effects in Written Discourse*. Ipskamp, Nijmegen, The Netherlands.

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

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

Cortes, Corinna and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Davidov, Dmitry, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in Twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Uppsala.

Ekman, Paul. 1992. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200.

Frank, Jane. 1990. You call that a rhetorical question?: Forms and functions of rhetorical questions in conversation. *Journal of Pragmatics*, 14(5):723–738.

Ghosh, Aniruddha 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*, pages 161–169, San Diego, CA.

Ghosh, Aniruddha 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*, pages 482–491, Copenhagen.

Ghosh, Debanjan, R. Alexander Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 186–196, Saarbrücken.

Ghosh, Debanjan, 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*, pages 1003–1012, Lisbon.

Ghosh, Debanjan and Smaranda Muresan. 2018. "With 1 follower I must be AWESOME : P." Exploring the role of irony markers in irony recognition. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018*, pages 588–591, Stanford, CA.

Gimpel, Kevin, 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, Portland, OR.

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

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

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

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

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, WA.

Huang, Yu-Hsiang, Hen-Hsen Huang, and Hsin-Hsi Chen. 2017. Irony detection with attentive recurrent neural networks. In *European Conference on Information Retrieval*, pages 534–540, Aberdeen.

Joshi, Aditya, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys*, 50(5):73:1–73:22.

Joshi, Aditya, 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)*, pages 757–762, Beijing.

Joshi, Aditya, Vaibhav Tripathi, Pushpak Bhattacharyya, and Mark J. Carman. 2016a. Harnessing sequence labeling for sarcasm detection in dialogue from TV series 'Friends.' In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 146–155, Berlin.

Joshi, Aditya, Vaibhav Tripathi, Kevin Patel, Pushpak Bhattacharyya, and Mark Carman. 2016b. Are word embedding-based features useful for sarcasm detection? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1006–1011, Austin, TX.

Justo, Raquel, Thomas Corcoran, Stephanie M. Lukin, Marilyn Walker, and M. Inés Torres. 2014. Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web. *Knowledge-Based Systems*, 69:124–133.

Khattri, Anupam, Aditya Joshi, Pushpak Bhattacharyya, and Mark Carman. 2015. Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 25–30, Lisboa.

Khodak, Mikhail, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the 11th edition of the Language Resources and Evaluation Conference (LREC 2018)*, pages 641–646, Miyazaki.

Krippendorff, Klaus. 2012. *Content Analysis: An Introduction to Its Methodology*. Sage.

Liebrecht, Christine, Florian Kunneman, and Antal Van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 29–37, Atlanta, GA.

Liu, Peng, Wei Chen, Gaoyan Ou, Tengjiao Wang, Dongqing Yang, and Kai Lei. 2014. Sarcasm detection in social media based on imbalanced classification. In *International Conference on Web-Age Information Management*, pages 459–471.

Maynard, Diana and Mark Greenwood. 2014. Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 4238–4243, Reykjavik.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Mishra, Abhijit, 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*, pages 1095–1104, Berlin.

Muresan, Smaranda, 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*, 67(11):2725–2737.

Oraby, Shereen, Vrindavan Harrison, Amita Misra, Ellen Riloff, and Marilyn Walker. 2017. Are you serious?: Rhetorical questions and sarcasm in social media dialog. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 310–319, Saarbrücken.

Oraby, Shereen, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41, Los Angeles, CA.

Parikh, Ankur, 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*, pages 2249–2255, Austin, TX.

Pennebaker, James W., Roger J. Booth, R. L. Boyd, and Martha E. Francis. 2015. *Linguistic inquiry and word count: LIWC 2015*. Austin, TX: Pennebaker Conglomerates (www.LIWC.net).

Ptáček, Tomáš, 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, Dublin.

Rajadesingan, Ashwin, 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*, pages 97–106, Shanghai.

Riloff, Ellen, 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*, pages 704–714, Seattle, WA.

Rocktäschel, Tim, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*, San Juan.

Schifanella, Rossano, Paloma de Juan, Joel Tetreault, and Liangliang Cao. 2016. Detecting sarcasm in multimodal social platforms. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1136–1145, Amsterdam.

Sha, Lei, Baobao Chang, Zhifang Sui, and Sujian Li. 2016. Reading and thinking: Re-read LSTM unit for textual entailment recognition. In *COLING*, pages 2870–2879.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Strapparava, Carlo and Alessandro Valitutti. 2004. Wordnet affect: An affective extension of Wordnet. In *LREC*, pages 1083–1086, Lisbon.

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

Tai, Kai Sheng, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing.

Tchokni, Simo, 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*, pages 485–494, Ann Arbor, MI.

Veale, Tony and Yanfen Hao. 2010. Detecting ironic intent in creative comparisons. In *European Conference on Artificial Intelligence*, 215:765–770.

Vinyals, Oriol, 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, Montreal.

Walker, Marilyn A., Jean E. Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *LREC*, pages 812–817, Istanbul.

Wallace, Byron C., 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, Baltimore, MD.

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

Wilson, Theresa, 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*, pages 347–354, Vancouver.

Xu, Kelvin, 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, Lille.
Yang, Zichao, 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, San Diego, CA.
Yin, Wenpeng, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272.
Zaremba, Wojciech, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.
Zayats, Victoria and Mari Ostendorf. 2018. Conversation modeling on Reddit using a graph-structured LSTM. *Transactions of the Association for Computational Linguistics*, 6:121–132.