Reactive Supervision: A New Method for Collecting Sarcasm Data

Boaz Shmueli[1,2,3,*], **Lun-Wei Ku**[2] and **Soumya Ray**[3]
[1]Social Networks and Human-Centered Computing, Taiwan International Graduate Program
[2]Institute of Information Science, Academia Sinica
[3]Institute of Service Science, National Tsing Hua University

Abstract

Sarcasm detection is an important task in affective computing, requiring large amounts of labeled data. We introduce **reactive supervision**, a novel data collection method that utilizes the dynamics of online conversations to overcome the limitations of existing data collection techniques. We use the new method to create and release a first-of-its-kind large dataset of tweets with sarcasm perspective labels and new contextual features. The dataset is expected to advance sarcasm detection research. Our method can be adapted to other affective computing domains, thus opening up new research opportunities.

1 Introduction

Sarcasm is ubiquitous in human conversations. As a form of insincere speech, the intent behind a sarcastic utterance is integral to its meaning. Perceiving a sarcastic utterance as genuine will often result in a complete reversal of the intended meaning, and vice versa (Gibbs, 1986). It is therefore crucial for affective computing systems and tasks, such as sentiment analysis and dialogue systems, to automatically detect sarcasm from the perspective of the author as well as the reader in order to avoid misunderstandings. Oprea and Magdy (2019) recently pioneered the study of *intended sarcasm* (by the author) vs. *perceived sarcasm* (by the reader) in the context of sarcasm detection tasks. The training of models for these tasks requires large amounts of labeled sarcasm data, with Twitter becoming a major source due to its popularity as a social network as well as the huge amounts of conversational text its users generate. Previous works describe three methods for collecting sarcasm data: distant supervision, manual annotation, and manual collection.

Distant supervision automatically collects "in-the-wild" sarcastic tweets by leveraging author-generated labels such as the #sarcasm hashtag (Davidov et al., 2010; Ptáček et al., 2014). This method generates large amounts of data at low cost, but labels are often noisy and biased (Bamman and Smith, 2015).

To improve quality, **manual annotation** asks humans to label given tweets as sarcastic or not. Since finding sarcasm in a large corpus is "a needle-in-a-haystack problem" (Liebrecht et al., 2013), manual annotation can be combined with distant supervision (Riloff et al., 2013). Still, low inter-annotator reliability is often reported (Swanson et al., 2014), resulting not only from the subjective nature of sarcasm but also the lack of cultural context (Joshi et al., 2016). Moreover, neither method collects both sarcasm perspectives: distant supervision collects intended sarcasm, while manual annotation can only collect perceived sarcasm.

Lastly, in **manual collection**, humans are asked to gather and report sarcastic texts, either their own (Oprea and Magdy, 2020) or by others (Filatova, 2012). However, both manual methods are slower and more expensive than distant supervision, resulting in smaller datasets.

To overcome the above limitations, we propose **reactive supervision**, a novel conversation-based method that offers automated, high-volume, "in-the-wild" collection of high-quality intended and perceived sarcasm data. We use our method to create and release the SPIRS sarcasm dataset[1].

2 Reactive Supervision

Reactive supervision exploits the frequent use in online conversations of a **cue tweet** — a reply that highlights sarcasm in a prior tweet. Figure 1 (left panel) shows a typical exchange on Twitter: C posts a sarcastic tweet. Unaware of C's sarcastic intent, B replies with an oblivious tweet. Lastly, A

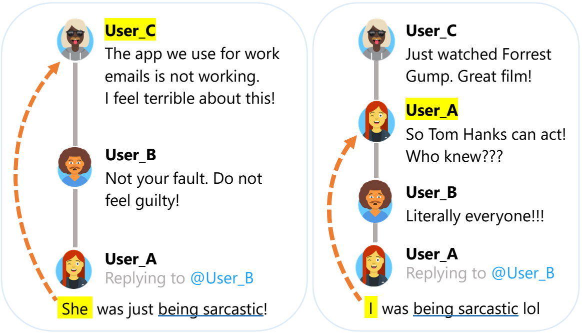
[*] Corresponding author: shmueli@iis.sinica.edu.tw

[1] github.com/bshmueli/SPIRS

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2553–2559,
November 16–20, 2020. ©2020 Association for Computational Linguistics

Person	Example Cue	Regular Expression	Example Author Sequences
1st	*I was only being sarcastic lol*	`^A[^A]*(A)[^A]*$`	$AB\mathbf{A}$, $AB\mathbf{A}C$, $AB\mathbf{A}B$
2nd	*Why are you being sarcastic?*	`^AA*(B)A*$`	$A\mathbf{B}$, $A\mathbf{B}A$, $A\mathbf{B}AA$
3rd	*She was just being sarcastic!*	`^AA*B[AB]*(C)[AB]*$`	$AB\mathbf{C}$, $AB\mathbf{C}B$, $ABA\mathbf{C}$

Table 1: The three grammatical person classes, with example cue tweets, corresponding regular expressions, and examples of matching author sequences. The bold author letter corresponds to the position of the sarcastic tweet.



User_C
The app we use for work emails is not working. I feel terrible about this!

User_B
Not your fault. Do not feel guilty!

User_A
Replying to @User_B
She was just being sarcastic!

User_C
Just watched Forrest Gump. Great film!

User_A
So Tom Hanks can act! Who knew???

User_B
Literally everyone!!!

User_A
Replying to @User_B
I was being sarcastic lol

Figure 1: Conversation threads. Left panel: 3rd-person cue with author sequence ABC. Right panel: 1st-person cue with author sequence $ABAC$.

alerts B by replying with a cue tweet (*She was just being sarcastic!*). Since A replies to B but refers to the sarcastic author in the 3rd person (*She*), C is necessarily the author of the perceived sarcastic tweet. Similarly, Figure 1 (right panel) shows how a 1st person cue (*I was just being sarcastic!*) can be used to unequivocally label intended sarcasm.

To capture sarcastic tweets, we thus first search for cue tweets (using the query phrase "being sarcastic", often used in responses to sarcastic tweets), then carefully examine each cue tweet to identify the corresponding sarcastic tweet.

The following formalizes our method.

2.1 Method

Definitions We define a thread to be a sequence of tweets $\{t_n, t_{n-1}, \ldots, t_1\}$, where t_{i+1} is a reply to t_i, $i = 1, \ldots, n-1$. Tweets are listed in reverse chronological order, with t_1 being the root tweet. The corresponding author sequence is $a_n a_{n-1} \ldots a_1$, were we replace the original author names with consecutive capital letters $(A, B, C, ...)$, starting with $a_n = A$. For example, Figure 1 (right panel) depicts a thread of length $n = 4$ with author sequence $ABAC$. Here $a_4 = a_2 = A$, $a_3 = B$, and $a_1 = C$ is the author of the root tweet.

Algorithm Given a thread $\{t_n, t_{n-1}, \ldots, t_1\}$ with cue tweet t_n by $a_n = A$, our aim is to identify the sarcastic tweet among $\{t_{n-1}, \ldots, t_1\}$. We first examine the personal subject pronoun used in the cue (I, you, s/he) and map it to a grammatical person class (1st, 2nd, 3rd). This informs us whether the sarcastic author is also the author of the cue (1st), its addressee (2nd), or another party (3rd). For each person class we then apply a heuristic to identify the sarcastic tweet.

For example, for a 1st-person cue tweet (e.g., *I was just being sarcastic!*), the sarcastic tweet must also be authored by A. If the earlier tweets in T contain exactly one tweet from A, it is unambiguously the sarcastic tweet. Otherwise, if there are two or more earlier tweets from A (or none), the sarcastic tweet cannot be unambiguously pinpointed and the entire thread is discarded. We formalize this rule by requiring the author sequence to match the regular expression `/^A[^A]*(A)[^A]*$/`, where the capturing group **`(A)`** corresponds to the sarcastic tweet[2]. We are able to use regular expressions because we use a string of letters to represent the author sequence. 2nd- and 3rd-person cues produce corresponding rules and patterns. Table 1 lists the three person classes, corresponding regular expressions, and example author sequences.

2.2 Advantages

Additional Tweet Types Along with each sarcastic tweet, we collect the **oblivious tweet** (the unsuspecting reply to the sarcastic tweet) when available. As far as we know, this is the first work that identifies and collects oblivious texts, a new type of data that can improve research on the (mis)understanding of sarcasm, with applications such as automated assistive systems for people with emotional or cognitive disabilities. If the sarcastic tweet is a reply, we also capture the **eliciting tweet**, which is the tweet that evoked the sarcastic reply. We provide more details in Appendix A.

[2] We use Perl-Compatible Regular Expressions (PCRE).

Extraction of Semantic Relations Being able to identify the various tweets types (cue, oblivious, sarcastic, eliciting), reactive supervision can be understood more abstractly as capturing *semantic dependency relations* between *utterances*[3]. Reactive supervision can thus be useful in the context of discourse analysis.

Context-Aware Annotation Our method uses cues from thread participants, who therefore serve as *de facto* annotators. As participants are familiar with the conversation's context, we overcome some quality issues of using external annotators, who are often unfamiliar with the conversation context due to cultural and social gaps (Joshi et al., 2016).

Sarcasm Perspective Previous datasets contain either intended or perceived sarcasm, but not both (Oprea and Magdy, 2019). Our method identifies and labels *both* intended and perceived sarcasm *within the same data context*: by their essence, 1st-person cue tweets capture intended sarcasm, while 2nd- and 3rd-person cues capture perceived sarcasm. We label a tweet as perceived sarcasm when at least one reader perceives the tweet as sarcastic and posts a cue tweet. Detecting perceived sarcasm is useful, for example, for training algorithms that flag sensitive texts which might be (mis)perceived as sarcastic (even by a single reader).

Faster Data Collection We tested González-Ibáñez et al. (2011)'s distant supervision method of collecting tweets ending with #sarcasm and related hashtags, fetching 171 tweets/day on average. During the same period, our method collected 312 tweets/day on average, an 82% rate improvement.

Summary of Advantages Table 2 summarizes the advantages of our best-of-all-worlds method over other approaches. Reactive supervision offers automated, in-the-wild, and context-aware detection of intended and perceived sarcasm data.

Method → **Feature ↓**	Distant Supervision	Manual Annotation	Manual Collection	**Reactive** **Supervision**
Automatic	✓	✗	✗	✔
In-the-wild	✓	✗	✗	✔
Oblivious Tweet	✗	✗	✗	✔
Context-Aware	✓	Maybe	Maybe	✔
Perspective	Intended	Perceived	Either	**Both**
Samples/Day	171	Manual	Manual	**312**

Table 2: Comparison of data collection methods.

Algorithm 1: Data collection pipeline.

```
Result: Set $S$ of Sarcastic Tweets
$S \leftarrow \{\}$
$candidates$ ← Fetch('being sarcastic')
for $cue$ in $candidates$ do
    switch Classify($cue$) do
        case 1st person do
            regexp ← ^A[^A]*(A)[^A]*$
        case 2nd person do
            regexp ← ^AA*(B)A*$
        case 3rd person do
            regexp ← ^AA*B[AB]*(C)[AB]*$
        case unknown do
            continue
    end
    $\{t_n (= cue), t_{n-1}, \ldots, t_1\}$ ← Traverse($cue$)
    $a_n a_{n-1} \ldots a_1$ ← authors($\{t_n, t_{n-1}, \ldots, t_1\}$)
    if $i$ ← Match(regexp, $a_n a_{n-1} \ldots a_1$) then
        $S \leftarrow S \cup \{t_i\}$
    end
end
```

3 SPIRS Dataset

We implemented reactive supervision using a 4-step pipeline (see Algorithm 1):

1. **Fetch** calls the Twitter Search API to collect cue tweets, using "being sarcastic" as the query.

2. **Classify** is a rule-based, precision-oriented classifier that classifies cues as 1st-, 2nd-, or 3rd-person according to the referred pronoun (I, you, s/he). If the cue cannot be accurately classified (e.g., a pronoun cannot be found, the cue contains multiple pronouns, negation words are present), the cue is classified as *unknown* and discarded.

3. **Traverse** calls the Twitter Lookup API to retrieve the thread by starting from the cue tweet and repeatedly fetching the parent tweet up to the root tweet.

4. Finally, **Match** matches the thread's author sequence with the corresponding regular expression. Unmatched sequences are discarded. Otherwise, the sarcastic tweet is identified and saved along with the cue tweet, as well as the eliciting and oblivious tweets when available.

The pipeline collected 65K cue tweets containing the phrase "being sarcastic" and corresponding threads during 48 days in October and November 2019. 77% of the cues were classified as *unknown* and discarded, ending with 15 000 English sarcastic tweets. In addition, 10 648 oblivious and 9 156 eliciting tweets were automatically captured. Table 3 summarizes the SPIRS dataset. We added 15 000 negative instances by sampling random English tweets captured during the same period, discarding tweets with sarcasm-related words or hashtags.

[3] It is worth noting that Hearst (1992) uses patterns to automatically extract *lexical relations* between *words*.

		# Tweets		
Person	**Perspective**	**Sarcastic**	**Oblivious**	**Eliciting**
1st	Intended	10 300	9 065	8 075
2nd	Perceived	3 000	—	842
3rd	Perceived	1 700	1583	239
Total		**15 000**	**10 648**	**9 156**

Table 3: SPIRS data breakdown by person class.

Sarcastic tweets can be either root tweets or replies. We found that the majority of intended sarcasm tweets are replies (78.4%), while the majority of perceived sarcasm tweets are root tweets (77.0%). Further dataset statistics on author sequence and tweet position distributions are available in Appendices B and C.

Reliability To assess our method's reliability in capturing sarcastic tweets, we manually inspected 200 random sarcastic tweets, along with their cue tweets, from each person class. The accuracy of sarcastic tweet labeling was high: 98.5%, 98%, and 97% for 1st-, 2nd-, and 3rd-person cue tweets, respectively. Table 4 shows samples of correct and incorrect cue tweet classifications.

Cue Tweet	**Pers.**	**Correct?**
Shudda been more clear...I was being sarcastic	1st	✓
I'm almost always being sarcastic, but this was real	1st	✗
Take it you are being sarcastic	2nd	✓
You do realize @user was being sarcastic right?	2nd	✗
She was being sarcastic. You missed the joke	3rd	✓
Mind blown. Had no idea he was being sarcastic	3rd	✗

Table 4: Correctly and incorrectly classified cue tweets.

4 Experiments and Analysis

We present dataset baselines for three tasks: sarcasm detection, sarcasm detection with conversation context, and sarcasm perspective classification, a new task enabled by our dataset.

4.1 Sarcasm Detection

The first experiment is sarcasm detection. We trained a total of three models: CNN (100 filters with a kernel size 3) and BiLSTM (100 units), both max-pooled and Adam-optimized with a learning rate of 0.0005; data was preprocessed as described in Tay et al. (2018); the embedding layer was preloaded with GloVe embeddings (Twitter data, 100 dimensions) (Pennington et al., 2014). We also fine-tuned a pre-trained base uncased BERT model (Devlin et al., 2019). For all three models, we used 5-fold cross-validation for training, holding out 20% of the data for testing.

Results are shown in Table 5 (top panel). BERT is the best performing model, with 70.3% accuracy. We compared SPIRS's classification results to the Ptáček et al. (2014) dataset, commonly used in sarcasm benchmarks. We found that Ptáček's accuracy is significantly higher (86.6%). We posit that it is because sarcasm is confounded with locale in the Ptáček (sarcastic tweets are from worldwide users; non-sarcastic tweets are from users near Prague), and thus classifiers learn features correlated to locale. We tested our hypothesis by replacing our negative samples with Ptáček's, which indeed resulted in boosting the accuracy by 19.1%.

4.2 Detection with Conversation Context

Our second sarcasm classification experiment uses conversation context by adding eliciting and oblivious tweets to the model. As far as we know, this is the first sarcasm-related task that uses oblivious texts. Our model concatenated the outputs of three identical 100-unit BiLSTMs (one per tweet: sarcastic, oblivious, eliciting) before feeding it into dense layers for classification. Tweets without surrounding context were not used in this task. Results are shown in Table 5 (middle panel). Accuracy for the full-context model was 74.7% (MCC 0.398).

Ablation Study We conducted context ablation experiments to identify the contribution of each tweet type. We found that removing the eliciting tweets reduces accuracy by 0.5% and MCC by 0.026. Removing the oblivious tweets, however, lowered accuracy by 3.4% to 71.4%, and the MCC dropped significantly by 31%, from 0.398 to 0.275. This illustrates the importance of the new oblivious text data provided in the dataset and suggests its usefulness in sarcasm-related tasks.

4.3 Perspective Classification

Taking advantage of the new labels in our dataset, we propose a new task to classify a sarcastic text's perspective: intended vs. perceived. Our results are displayed in Table 5 (bottom panel), demonstrating the superiority of BERT over the other models, with an accuracy of 68.2% and MCC of 0.366.

Error Analysis We carefully examined the errors to analyze the causes of perspective misclassification. We observed that misclassified-as-intended tweets (e.g., "You're lost!", "Omg that was so

Task	Dataset	Model	P	R	F1	Acc	MCC
Sarcasm	SPIRS	CNN	67.2 (1.8)	73.6 (5.1)	65.0 (1.2)	65.8 (0.5)	0.308 (0.011)
Detection	(our dataset)	BiLSTM	68.9 (2.1)	75.4 (5.5)	67.1 (0.9)	67.9 (0.3)	0.350 (0.008)
	N=19 384	BERT	**70.1 (1.1)**	**77.4 (1.2)**	**69.9 (0.5)**	**70.3 (0.5)**	**0.402 (0.008)**
	Ptáček	CNN	79.1 (0.8)	87.5 (1.3)	77.9 (0.6)	79.2 (0.6)	0.566 (0.012)
	N=49 766	BiLSTM	82.4 (1.6)	87.6 (2.9)	80.9 (0.1)	81.7 (0.2)	0.622 (0.002)
		BERT	**87.0 (0.6)**	**90.9 (0.6)**	**86.0 (0.2)**	**86.6 (0.2)**	**0.721 (0.004)**
	Ptáček (−)	CNN	84.3 (1.6)	82.6 (2.5)	83.6 (0.8)	83.6 (0.8)	0.673 (0.017)
	SPIRS (+)	BiLSTM	86.2 (2.8)	86.7 (2.8)	86.4 (0.7)	86.4 (0.7)	0.729 (0.012)
	N=21 138*	BERT	**89.8 (0.7)**	**89.1 (0.7)**	**89.4 (0.2)**	**89.4 (0.2)**	**0.788 (0.004)**
Sarcasm	SPIRS	3 X BiLSTM	**77.7 (1.1)**	**87.9 (3.5)**	**68.9 (0.7)**	**74.8 (0.6)**	**0.398 (0.007)**
Detection	(our dataset)	w/o eliciting	75.6 (1.1)	91.4 (2.8)	66.3 (1.4)	74.3 (0.3)	0.372 (0.005)
w/ Conversation	N=7 810*	w/o oblivious	72.4 (2.4)	93.3 (4.5)	58.8 (6.2)	71.4 (1.4)	0.275 (0.053)
Context		w/o both	73.2 (2.7)	90.8 (6.6)	60.3 (4.6)	71.2 (0.4)	0.282 (0.033)
Sarcasm	SPIRS	CNN	65.5 (1.2)	61.7 (3.3)	64.4 (0.5)	64.5 (0.5)	0.291 (0.009)
Perspective	(our dataset)	BiLSTM	66.8 (2.3)	63.1 (5.8)	65.5 (0.7)	65.6 (0.7)	0.315 (0.015)
Classification	N=6 324*	BERT	**70.0 (2.9)**	**63.8 (5.7)**	**68.0 (1.7)**	**68.2 (1.6)**	**0.366 (0.032)**

Table 5: Baselines. We report precision, recall, macro-F1, accuracy, and MCC (Matthews correlation coefficient). Mean and standard deviation were calculated using 5-fold cross-validation. N is the number of instances after preprocessing. *Dataset classes were balanced using majority class downsampling.

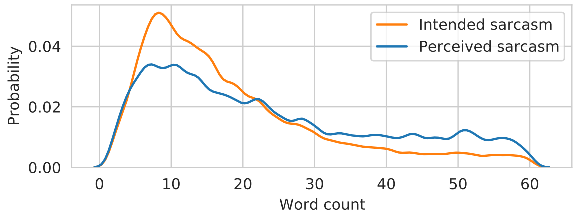


Figure 2: Word count distribution in SPIRS

funny”) had, on average, almost half the word count of misclassified-as-perceived tweets (17.2 vs. 27.8). We posit that longer, more informative texts make sarcasm easier to perceive; hence, short perceived sarcasm or long intended sarcasm might introduce errors. Analysis of the dataset’s word count distribution supports our hypothesis (see Figure 2).

Looking for further error sources, we inspected *short* intended tweets that were misclassified, for example “great friends i have!” and “My mom is so beautiful”. These tweets can be read as root tweets and not as replies, yet most intended sarcasm tweets are replies while most perceived sarcasm tweets are root tweets (see Section 3). We hypothesize that the classifier learns discourse-related features (original tweet vs. reply tweet), which can lead to these errors. Further analysis of sarcasm perspective and its interplay with sarcasm pragmatics is a promising avenue for future research.

5 Conclusion

We present an innovative method for collecting sarcasm data that exploits the natural dynamics of online conversations. Our approach has multiple advantages over all existing methods. We used it to create and release SPIRS, a large sarcasm dataset with multiple novel features. These new features, including labels for sarcasm perspective and unique context (e.g., oblivious texts), offer opportunities for advances in sarcasm detection.

Reactive supervision is generalizable. By modifying the cue tweet selection criteria, our method can be adapted to related domains such as sentiment analysis and emotion detection, thereby advancing the quality and quantity of data collection and offering new research directions in affective computing.

Acknowledgements

This research was partially supported by the Ministry of Science and Technology of Taiwan under contracts MOST 108-2221-E-001-012-MY3 and MOST 108-2321-B-009-006-MY2.

References

David Bamman and Noah A Smith. 2015. Contextualized Sarcasm Detection on Twitter. In *Ninth International AAAI Conference on Web and Social Media*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised Recognition of Sarcastic Sentences in Twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 107–116, Stroudsburg, PA, USA. Association for Computational Linguistics. Event-place: Uppsala, Sweden.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Elena Filatova. 2012. Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 392–398, Istanbul, Turkey. European Language Resources Association (ELRA).

Raymond W Gibbs. 1986. On the psycholinguistics of sarcasm. *Journal of Experimental Psychology: General*, 115(1):3.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying Sarcasm in Twitter: A Closer Look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 581–586, Stroudsburg, PA, USA. Association for Computational Linguistics. Event-place: Portland, Oregon.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

Aditya Joshi, Pushpak Bhattacharyya, Mark Carman, Jaya Saraswati, and Rajita Shukla. 2016. How Do Cultural Differences Impact the Quality of Sarcasm Annotation?: A Case Study of Indian Annotators and American Text. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 95–99, Berlin, Germany. Association for Computational Linguistics.

Christine Liebrecht, 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, Georgia. Association for Computational Linguistics.

Silviu Oprea and Walid Magdy. 2019. Exploring author context for detecting intended vs perceived sarcasm. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2859, Florence, Italy. Association for Computational Linguistics.

Silviu Oprea and Walid Magdy. 2020. iSarcasm: A Dataset of Intended Sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm Detection on Czech and English Twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. Association for Computational Linguistics.

Reid Swanson, Stephanie Lukin, Luke Eisenberg, Thomas Corcoran, and Marilyn Walker. 2014. Getting reliable annotations for sarcasm in online dialogues. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4250–4257, Reykjavik, Iceland. European Language Resources Association (ELRA).

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Reasoning with Sarcasm by Reading In-Between. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1010–1020, Melbourne, Australia. Association for Computational Linguistics.

A Search Pattern Production

We construct the regular expression for capturing all tweet types — sarcastic, oblivious, and eliciting — given a 3rd-person cue tweet. Similar logic produces the patterns for 1st- and 2nd-person cues.

The cue tweet author (A) refers to the sarcastic tweet author in the 3rd person (e.g., *She was being sarcastic!*); we thus assume that A's tweet is a response to a second author B, but refers to a third author C (the sarcastic author). To unambiguously pinpoint the sarcastic tweet, C can only appear once in the author sequence. Moreover, only A, B, and C can participate in the thread. Finally, C's tweet can either be a root tweet or a reply to another tweet. The combination of these constraints leads to the regular expression `/^(A)(A*B[AB]*)(C)([AB]*)$/`.

`(A)` is the cue tweet. `(A*B[AB]*)` forces at least one tweet from B (to which A responded). `(C)` is the sarcastic tweet. Finally, `([AB]*)` represents optional tweets from A or B. If the author sequence matches the regular expression, we can unambiguously identify the sarcastic author and the corresponding sarcastic tweet. We also use the search pattern to find the oblivious and eliciting tweets. We assume that the cue tweet `(A)` is triggered by an oblivious tweet from B. Thus, if `(A*B[AB]*)` contains *exactly* one B, we designate the corresponding tweet as oblivious. Likewise, `([AB]*)` contains the eliciting tweet.

Table 6 lists the search patterns for the three person classes. Note that the 2nd-person pattern does not include an oblivious tweet because A's cue tweet is a response to a sarcastic tweet from B, i.e., it is not triggered by an oblivious tweet.

Person	Regular Expression
1st	`^(A)([^A]*)(A)([^A]*)$`
2nd	`^(A)A*(B)(A*)$`
3rd	`^(A)(A*B[AB]*)(C)([AB]*)$`

Table 6: Person classes and their search patterns. The capturing groups' colors correspond to the locations of the cue, oblivious, sarcastic and eliciting tweets.

B Author Sequence Distribution

Table 7 shows the most common author sequences in SPIRS. The different colors correspond to the different tweet types. The most common pattern for 1st-person cues is $ABAC$ (as in Figure 1, right panel). AB is the most common pattern for 2nd-person cues, which denote a sarcastic root tweet followed immediately by a cue tweet (e.g., *Why are you being sarcastic?*). For 3rd-person cues, the most common pattern is ABC (as in Figure 1, left panel). Note that some patterns appear in more than one person class. For example, ABA appears in both 1st- and 2nd-person classes, while $ABAC$ appears in both 1st- and 3rd-person.

		# Tweets		
Person	**Patterns**	**Sarcast.**	**Obliv.**	**Elicit.**
1st (Intended)	$ABAC$	2 841	2 841	2 841
	ABA	1 818	1 818	—
	$ABAB$	1 551	1 551	1 551
	Other	4 090	2 855	2 683
	Subtotal	**10 300**	**9 065**	**8 075**
2nd (Perceived)	AB	2 122	—	—
	ABA	782	—	782
	Other	96	—	60
	Subtotal	**3 000**	**—**	**842**
3rd (Perceived)	ABC	1 235	1 235	—
	$ABCB$	119	119	119
	$ABAC$	110	110	—
	Other	236	119	120
	Subtotal	**1 700**	**1 583**	**239**
	Total	**15 000**	**10 648**	**9 156**

Table 7: The most common author patterns by person class. The colors denote the locations of the cue, oblivious, sarcastic and eliciting tweets.

C Tweet Position Distribution

Reactive supervision enables the measurement of conversation position statistics for sarcastic tweets on Twitter. Given a thread $\{t_n, \ldots, t_i = s, \ldots, t_1\}$ with cue tweet t_n, sarcastic tweet $t_i = s$, and root tweet t_1, we define the *position* of the sarcastic tweet as the distance $i - 1$ between the sarcastic tweet and the root. Furthermore, the *cue lag* is the distance $n - i$ between the cue and the sarcastic tweet. Table 8 shows the distribution of sarcastic tweets by position and cue lag in the SPIRS dataset.

Root tweets (*position* = 0) account for 39% of sarcastic tweets. A further 39% of sarcastic tweets are direct replies to root tweets (*position* = 1). Interestingly, only 25% of cue tweets are direct replies to their sarcastic targets (*lag* = 1), while an overwhelming 71% have a lag of 2, mostly reflecting a response to an intermediate oblivious tweet. We further find that the average thread length is 3.9 tweets, while the average lag is 1.8 tweets.

	Distance from the root tweet						
Cue lag	0	1	2	3	4	5+	**Total**
1	16.5	7.2	0.9	0.3	0.1	0.2	25.1
2	20.6	30.6	11.4	3.8	1.7	2.3	70.4
3+	1.9	1.3	0.7	0.3	0.1	0.2	4.5
Total	**39.0**	**39.1**	**13.0**	**4.3**	**1.9**	**2.7**	**100.0**

Table 8: % of sarcastic tweets by position (distance from the root tweet) and cue lag.