Exploring Author Context for Detecting Intended vs Perceived Sarcasm

Silviu Vlad Oprea
School of Informatics
University of Edinburgh
Edinburgh, United Kingdom
silviu.oprea@ed.ac.uk

Walid Magdy
School of Informatics
University of Edinburgh
Edinburgh, United Kingdom
wmagdy@inf.ed.ac.uk

Abstract

We investigate the impact of using author context on textual sarcasm detection. We define author context as the embedded representation of their historical posts on Twitter and suggest neural models that extract these representations. We experiment with two tweet datasets, one labelled manually for sarcasm, and the other via tag-based distant supervision. We achieve state-of-the-art performance on the second dataset, but not on the one labelled manually, indicating a difference between intended sarcasm, captured by distant supervision, and perceived sarcasm, captured by manual labelling.

1 Introduction

Sarcasm is a form of irony that occurs when there is a discrepancy between the literal meaning of an utterance and its intended meaning. This discrepancy is used to express a form of dissociative attitude towards a previous proposition, often in the form of contempt or derogation (Wilson, 2006).

Sarcasm is omnipresent on the social web and can be highly disruptive of systems that harness this data (Maynard and Greenwood, 2014). It is therefore imperative to devise model for textual sarcasm detection. The effectiveness of such models depends on the quality of labelled data used for training. Two methods are commonly used to label texts for sarcasm: manual labelling by human annotators; and tag-based distant supervision. In the latter, texts are considered sarcastic if they contain specific tags, such as #sarcasm and #sarcastic.

Most work on computational sarcasm detection extracts lexical and pragmatic cues available in the text being classified (Campbell and Katz, 2012; Riloff et al., 2013; Joshi et al., 2016; Tay et al., 2018). However, sarcasm is a contextual phenomenon and detecting it often requires prior information about the author, audience and previous interactions between them, that originates beyond the text itself (Rockwell and Theriot, 2001a).

In this work we investigate the impact of author context on the current sarcastic behaviour of the author. We identify author context with the embedded representation of their historical tweets. We use the term *user* to refer to the author of a tweet and the phrase *user embedding* to refer to such a representation. Given a tweet t posted by user u^t with user embedding e^t, we address two questions: (1) Is e^t predictive of the sarcastic nature of t? (2) Is the predictive power of e^t on the sarcastic nature of t the same if t is labelled via manual labelling vs distant supervision?

To our knowledge, previous research that considers author context (Rajadesingan, Zafarani, and Liu, 2015; Bamman and Smith, 2015; Amir et al., 2016; Hazarika et al., 2018) only experiments on distant supervision datasets. We experiment on datasets representative of both labelling methods, namely Riloff (Riloff et al., 2013), labelled manually, and Ptacek (Ptáček, Habernal, and Hong, 2014), labelled via distant supervision.

We suggest neural models to build user embeddings and achieve state-of-the-art results on Ptacek, but not on Riloff. Comparing and analyzing the discrepancy, our findings indicate a difference between the sarcasm that is *intended* by the author, captured by distant supervision, represented in Ptacek, and sarcasm that is *perceived* by the audience, captured by manual labelling, represented in Riloff. This difference has been highlighted by linguistic and psycholinguistic studies in the past (Rockwell and Theriot, 2001b; Pexman, 2005), being attributed to socio-cultural differences between the author and the audience. However, up to our knowledge, it has not been considered in the context of sarcasm detection so far. Our work suggests a future research direction in sarcasm detection where the two types of sarcasm are treated as separate phenomena and socio-cultural

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2854–2859
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

differences are taken into account.

2 Background

2.1 Sarcasm Detection

Based on the information considered when classifying a text as sarcastic or non-sarcastic, we identify two classes of models across literature: local models and contextual models.

Local Models Local models only consider information available within the text being classified. Most work in this direction considers linguistic incongruity (Campbell and Katz, 2012) to be a marker of sarcasm. Riloff et al. (2013) consider a positive verb used in a negative sentiment context to indicate sarcasm. Joshi et al. (2016) use the cosine similarity between embedded representations of words. Recent work attempts to capture incongruity using a neural network with an intra-attention mechanism (Tay et al., 2018).

Contextual Models Contextual models utilize both local and contextual information. There is a limited amount of work in this direction. Wallace, Choe, and Charniak (2015), working with Reddit data, include information about the forum type where the post to be classified was posted. For Twitter data, Rajadesingan, Zafarani, and Liu (2015) and Bamman and Smith (2015) represent user context by a set of manually-curated features extracted from their historical tweets. Amir et al. (2016) merge all historical tweets of a user into one historical document and use the Paragraph Vector model (Le and Mikolov, 2014) to build a representation of that document. Building on their work, Hazarika et al. (2018) extract in addition personality features from the historical document. Despite reporting encouraging results, these models are only tested on datasets labelled via distant supervision. In our work, we compare the performance of our models when tested on datasets representative of both manual annotation and distant supervision.

2.2 Intended vs Perceived Sarcasm

Dress et al. (2008) notice a lack of consistence in how sarcasm is defined by people of different socio-cultural backgrounds. As a result, an utterance that is intended as sarcastic by its author might not be perceived as such by audiences of different backgrounds (Rockwell and Theriot, 2001a). When a tweet is sarcastic from the perspective of its author, we call the resulting phenomenon *intended sarcasm*. When it is sarcastic from the perspective of an audience member, we call the phenomenon *perceived sarcasm*.

3 Sarcasm Datasets

We test our models on two popular tweet datasets, one labelled manually and the other via distant supervision.

3.1 Riloff dataset

The Riloff dataset consists of 3,200 tweet IDs. These tweets were manually labeled by third party annotators. The labels capture the subjective perception of the annotators (perceived sarcasm). Three separate labels were collected for each tweet and the dominant one was chosen as the final label.

We attempted to collect the corresponding tweets using the Twitter API[1], as well as the historical timeline tweets for each user, to be used later for building user embeddings. For a user with tweet t in Riloff, we collected those historical tweets posted before t. Only 701 original tweets, along with the corresponding user timelines, could be retrieved. Others have either been removed from Twitter, the corresponding user accounts have been disabled, or the API did not retrieve any historical tweets.

Table 1 shows the label distribution across this dataset. We divided the dataset into ten buckets, using eight for training, one for validation and one for testing. The division into buckets is stratified by users, i.e. all tweets from a user end up in the same bucket. Stratification makes sure any specific embedding is only used during training, during validation, or during testing. We further ensured the overall class balance is represented in all of the three sets. Table 1 shows the size of each set.

3.2 Ptacek dataset

The Ptacek dataset consists of 50,000 tweet IDs labelled via distant supervision. Tags used as markers of sarcasm are #sarcasm, #sarcastic, #satire and #irony. This dataset reflects intended sarcasm, since the original poster tagged their own tweet as sarcastic through the hashtag.

In a similar setting as with Riloff we could only collect 27,177 tweets and corresponding time-

[1] https://developer.twitter.com

dataset	size	sarcastic	non-sarcastic	train	valid	test
Riloff	701	192	509	551	88	62
Ptacek	27,177	15,164	12,013	21,670	2,711	2,797

Table 1: Label distribution across our datasets; and distribution into train, validation and test sets.

lines. We divided them into ten buckets and stratified by users. During preprocessing we removed all sarcasm-marking tags from both the training tweets and the historical tweets. Table 1 shows statistics on both datasets.

4 Contextual Sarcasm Detection Models

Let T be a set of tweets. For any $t \in T$, let u^t be the user who posted tweet t. Let h^t be a set of historical tweets of user u^t, posted before t, with $h^t \cap T = \emptyset$ and let e^t be the embedding of user u^t, i.e. a vector representation of h^t. Let $Y = \{\text{sarcastic, non-sarcastic}\}$ be the output space. Our goal is to find a model $m : \{(t, e^t)|t \in T\} \rightarrow Y$.

As a baseline, we implement the SIARN (Single-Dimension Intra-Attention Network) model proposed by (Tay et al., 2018), since it achieves the best published results on both our datasets. SIARN only looks at the tweet being classified, that is $\text{SIARN}(t, e^t) = m'(t)$.

Further, we introduce two classes of models: exclusive and inclusive models. In exclusive models, the decision whether $t \in T$ is sarcastic or not is independent of t, i.e. $m(t, e^t) = m'(e^t)$. The content of the tweet being classified is not considered, prediction being based solely on user historical tweets. The architecture of such a model is shown in Figure 1. We feed the user embedding e^t to a layer with softmax activations to output a probability distribution over Y. We name these models EX-[emb], where [emb] is the name of the user embedding model.

Inclusive models account for both t and e^t, as shown in Figure 1. We start with the feature vector f^t extracted by SIARN from t. We then concatenate f^t with e^t and use an output layer with softmax activations. We name these models IN-[emb], where [emb] is the user embedding model. We now look at several user embedding models that build e^t for a user u^t as a representation of h^t. Recall that $\forall u \in usr(T) : hist(u) \cap T = \emptyset$, where $usr(T)$ is the image of T under usr.

CASCADE Embeddings Up to our knowledge, the user embedding model that has proven most informative in a sarcasm detection pipeline so far is CASCADE (Hazarika et al., 2018). However, it has only been tested on a dataset of Reddit[2] posts labelled via distant supervision. We test it on our datasets. Following original authors, we merge all tweets from h^t in a single document d^t, giving corpus $C = \{d^t|t \in T\}$. Using the Paragraph Vector model (Le and Mikolov, 2014) we generate a representation v^t of d^t. Next, we feed d^t to a neural network pre-trained on the personality detection corpus released by Matthews and Gilliland (1999), which contains labels for the Big-Five personality traits (Goldberg, 1993). We merge the resulting hidden state p^t of the network with v^t using Generalized Canonical Correlation Analysis (GCCA) as described by Hazarika et al. (2018) to get e^t.

W-CASCADE Embeddings CASCADE treats all historical tweets in the same manner. However, as studies in cognitive psychology argue (Kellogg, 2001), long-term working memory plays an important role in verbal reasoning and textual comprehension. We therefore expect recent historical tweets to have a greater influence on the current behaviour of a user, compared to older ones. To account for this, we suggest the following model that accounts for the temporal arrangement of historical tweets. We first use CASCADE to build v_r^t and p_r^t, and to merge them into e_r^t using GCCA, $\forall r \in h^t$. We then divide the sequence $\langle e_{r_1}^t, e_{r_2}^t, \ldots, e_{r_{|h^t|}}^t \rangle$ into ten contiguous partitions and multiply each vector with the index of the partition it belongs to. That is, we multiply $e_{r_i}^t$ by $i \ \% \ |h^t| + 1$, where $\%$ is the modulus operator. By convention, the tweet with the highest index is the most recent one. Finally, we sum the resulting vectors and normalize the result to get e^t.

ED Embeddings One of the main advantages of the encoder-decoder model (Sutskever, Vinyals, and Le, 2014), commonly used for sequence prediction tasks, is its ability to handle inputs and outputs of variable length. The encoder, a recurrent network, transforms an input sequence into an internal representation of fixed dimension. The decoder, another recurrent network, generates an

[2]https://www.reddit.com

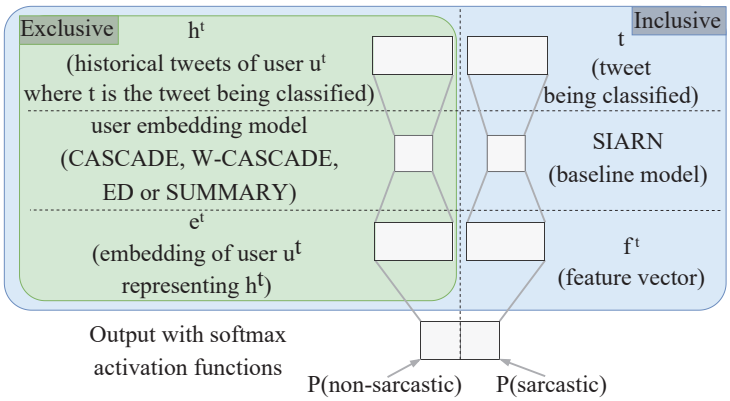

Figure 1: The architecture of the models used. Exclusive models do not use the current tweet being classified, prediction being based solely on user history. Inclusive models use both user history and the current tweet.

output sequence using this representation. We use bi-directional LSTM cells (Schuster and Paliwal, 1997) and identify $e^t_{r_i}$, $1 \leq i \leq |h^t|$, with the internal state of the encoder after feeding in r_i. The training objective is to reconstruct the input r_i. We employ the same weighting technique as we did for W-CASCADE to construct e^t.

SUMMARY Embeddings We use an encoder-decoder model as in the previous paragraph, but change the objective from reconstructing the input to summarizing it. We pre-train the model on the Gigaword standard summarization corpus[3].

5 Effect of Context on Sarcasm Detection

5.1 Experimental Setup

We filter out all tweets shorter than three words and replace all words that only appear once in the entire corpus with an UNK token. Then, we encode each tweet as a sequence of word vectors initialized using GloVe embeddings (Pennington, Socher, and Manning, 2014). Following the authors SIARN, our baseline, we set the word embedding dimension to 100. We tune the dimension of all CASCADE embeddings to 100 on the validation set. For comparability, we set W-CASCADE embeddings to the same dimension. For CASCADE embeddings we make use of the implementation available at `https://github.com/SenticNet/cascade`. When training ED and SUMMARY, our decoder implements attention over the input vectors. We use the general global attention mechanism suggested by Luong, Pham, and Manning (2015). We implement both ED and SUMMARY using the OpenNMT toolkit (Klein et al., 2017).

For comparability with SIARN, our baseline, we follow its authors in setting a batch size of 16 for the Riloff dataset, and of 512 for the Ptacek dataset, and in training for 30 epochs using the RMSProp optimizer (Tieleman and Hinton, 2012) with a learning rate of 0.001. Our code and data can be obtained by contacting us.

[3] `https://github.com/harvardnlp/sent-summary`

	Model	Riloff	Ptacek
	SIARN (baseline)	0.711	0.863
exclusive	EX-CASCADE	0.457	0.802
	EX-W-CASCADE	0.478	**0.922**
	EX-ED	**0.546**	0.873
	EX-SUMMARY	0.492	0.845
inclusive	IN-CASCADE	0.723	0.873
	IN-W-CASCADE	0.714	**0.934**
	IN-ED	**0.739**	0.887
	IN-SUMMARY	0.679	0.892

Table 2: F1 score achieved on the Riloff and Ptacek datasets for both exclusive and inclusive models. Best results for each model class are highlighted in **bold**.

Model	Riloff	#Riloff
EX-CASCADE	0.457	0.818
EX-W-CASCADE	0.478	0.797
EX-ED	**0.545**	**0.827**
EX-SUMMARY	0.492	0.772

Table 3: F1 score achieved by the exclusive models on the #Riloff dataset, compared to Riloff dataset. Best results are highlighted in **bold**.

5.2 Results

All results are reported in Table 2. User embeddings show remarkable predictive power on the Ptacek dataset. In particular, using the EX-W-

	with tag	without any tag
labelled sarcastic	190	2
labelled non-sarcastic	217	292

Table 4: Disagreement between manual labels and the presence of sarcasm tags in the Riloff dataset, as discussed in Section 5.3.

CASCADE model, we get better results (f1-score 0.922) than the baseline (f1-score 0.863) without even looking at the tweet being predicted. On the Riloff dataset, however, user embeddings seem to be far less informative, with EX-W-CASCADE yielding an f1-score of only 0.478. Out of the exclusive models, we get the highest f1-score of 0.546 using EX-ED on Riloff. By contrast we get 0.873 on Ptacek using EX-ED.

The state-of-the-art performance of exclusive models on Ptacek indicate that users seem to have a prior disposition to being either sarcastic or non-sarcastic, which can be deduced from historical behaviour. However, this behaviour can change over time, as we achieve better performance when accounting for the temporal arrangement of historical tweets, as we do in W-CASCADE.

On the Riloff dataset the performance of exclusive models is considerably lower. In the following, we investigate the possible reasons for this large difference in performance between the two datasets.

5.3 Performance Analysis

Riloff dataset is annotated manually, which might not reflect the intention of the users, but rather the subjective perception of the annotators. In this light, we could expect user embeddings to have poor predictive power. Perhaps annotator embeddings would shed more light.

We noticed that many of the tweets in Riloff contain one or more of the tags that were used to mark sarcasm in Ptacek. For all tweets in Riloff, we checked the agreement between containing such a tag, and being manually annotated as sarcastic. The results are shown in Table 4. Note that the statistics shown are not for the entire dataset as published by Riloff et al. (2013), but for the subset of tweets coming from users without blocked profiles and from which we could gather historical tweets, as discussed in Section 3. We notice a large disagreement. In particular, 217 out of the 509 tweets that were annotated manually as non-sarcastic contained such a tag. The lack of coherence between the presence of sarcasm tags and manual annotations in the Riloff dataset suggests that the two labelling methods capture distinct phenomena, considering the subjective nature of sarcasm. Previous research in linguistics and psycholinguistics (Rockwell and Theriot, 2001b; Pexman, 2005) attributes this difference to socio-cultural differences between the author and the audience and shows that the difference persists even when contextual information is provided.

To investigate further, we re-labelled the Riloff dataset via distant supervision considering these tags as markers of sarcasm, to create the #Riloff dataset. We applied the exclusive models on #Riloff and noticed a considerably higher predictive power than on Riloff. Results are reported in Table 3. Author history seems therefore predictive of authorial sarcastic intention, but not of external perception. This could indicate that future work should differentiate between the two types of sarcasm: intended and perceived. Both are important to detect, for applications such as opinion mining for the former and hate speech detection for the latter.

6 Conclusion

We studied the predictive power of user embeddings in textual sarcasm detection across datasets labelled via both manual labelling and distant supervision. We suggested several neural models to build user embeddings, achieving state-of-the-art results for distant supervision, but not for manual labelling. We account for discrepancy by reference to the different type of sarcasm captured by the two labelling methods, attributed by previous research in linguistics and psycholinguistics (Rockwell and Theriot, 2001b; Pexman, 2005) to socio-cultural differences between the author and the audience. We suggest a future research direction in sarcasm detection where the two types of sarcasm are treated as separate phenomena and socio-cultural differences are taken into account.

7 Acknowledgements

This work was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1); the University of Edinburgh; and The Financial Times.

References

Amir, S.; Wallace, B. C.; Lyu, H.; Carvalho, P.; and Silva, M. J. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *CoNLL*, 167–177. ACL.

Bamman, D., and Smith, N. A. 2015. Contextualized sarcasm detection on twitter. In *ICWSM*, 574–577. AAAI Press.

Campbell, J. D., and Katz, A. N. 2012. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes* 49(6):459–480.

Dress, M. L.; Kreuz, R. J.; Link, K. E.; and Caucci, G. M. 2008. Regional variation in the use of sarcasm. *JLS* 27(1):71–85.

Goldberg, L. R. 1993. The structure of phenotypic personality traits. *American psychologist* 48(1):26.

Hazarika, D.; Poria, S.; Gorantla, S.; Cambria, E.; Zimmermann, R.; and Mihalcea, R. 2018. Cascade: Contextual sarcasm detection in online discussion forums. In *COLING*, 1837–1848. ACL.

Joshi, A.; Tripathi, V.; Patel, K.; Bhattacharyya, P.; and Carman, M. 2016. Are word embedding-based features useful for sarcasm detection? In *EMNLP*, 1006–1011. ACL.

Kellogg, R. T. 2001. Long-term working memory in text production. *Memory & Cognition* 29(1):43–52.

Klein, G.; Kim, Y.; Deng, Y.; Senellart, J.; and Rush, A. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *ACL*, 67–72. ACL.

Le, Q., and Mikolov, T. 2014. Distributed Representations of Sentences and Documents. In *ICML*, 1188–1196. PMLR.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP*, 1412–1421. ACL.

Matthews, G., and Gilliland, K. 1999. The personality theories of H.J. Eysenck and J.A. Gray: a comparative review. *Personality and Individual Differences* 26(4):583–626.

Maynard, D., and Greenwood, M. 2014. Who cares about sarcastic tweets? investigating the impact of sarcasm on sentiment analysis. In *LREC*. ELRA.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543. ACL.

Pexman, P. M. 2005. Social Factors in the Interpretation of Verbal Irony: The Roles of Speaker and Listener Characteristics.

Ptáček, T.; Habernal, I.; and Hong, J. 2014. Sarcasm detection on czech and english twitter. In *COLING*, 213–223. ACL.

Rajadesingan, A.; Zafarani, R.; and Liu, H. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *WSDM*, 97–106. ACM.

Riloff, E.; Qadir, A.; Surve, P.; De Silva, L.; Gilbert, N.; and Huang, R. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, 704–714. ACL.

Rockwell, P., and Theriot, E. M. 2001a. Culture, gender, and gender mix in encoders of sarcasm: A self-assessment analysis. *Communication Research Reports* 18(1):44–52.

Rockwell, P., and Theriot, E. M. 2001b. Culture, gender, and gender mix in encoders of sarcasm: A self-assessment analysis. *Communication Research Reports* 18(1):44–52.

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*, 3104–3112. MIT Press.

Tay, Y.; Luu, A. T.; Hui, S. C.; and Su, J. 2018. Reasoning with sarcasm by reading in-between. In *ACL*, 1010–1020. ACL.

Tieleman, T., and Hinton, G. 2012. Lecture 6.5—RMSProp: Divide the gradient by a running average of its recent magnitude. Coursera: Neural Networks for Machine Learning. Accessed: 2019-03-01.

Wallace, B. C.; Choe, D. K.; and Charniak, E. 2015. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *ACL-IJCNLP*, 1035–1044. ACL.

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