Deep contextualized word representations for detecting sarcasm and irony

Suzana Ilić[1], Edison Marrese-Taylor[2], Jorge A. Balazs[2], Yutaka Matsuo[2]
University of Innsbruck, Austria[1]
suzana.ilic@student.uibk.ac.at
Graduate School of Engineering, The University of Tokyo, Japan[2]
{emarrese,jorge,matsuo}@weblab.t.u-tokyo.ac.jp

Abstract

Predicting context-dependent and non-literal utterances like sarcastic and ironic expressions still remains a challenging task in NLP, as it goes beyond linguistic patterns, encompassing common sense and shared knowledge as crucial components. To capture complex morpho-syntactic features that can usually serve as indicators for irony or sarcasm across dynamic contexts, we propose a model that uses character-level vector representations of words, based on ELMo. We test our model on 7 different datasets derived from 3 different data sources, providing state-of-the-art performance in 6 of them, and otherwise offering competitive results.

1 Introduction

Sarcastic and ironic expressions are prevalent in social media and, due to the tendency to invert polarity, play an important role in the context of opinion mining, emotion recognition and sentiment analysis (Pang and Lee, 2006). Sarcasm and irony are two closely related linguistic phenomena, with the concept of meaning the opposite of what is literally expressed at its core. There is no consensus in academic research on the formal definition, both terms are non-static, depending on different factors such as context, domain and even region in some cases (Filatova, 2012).

In light of the general complexity of natural language, this presents a range of challenges, from the initial dataset design and annotation to computational methods and evaluation (Chaudhari and Chandankhede, 2017). The difficulties lie in capturing linguistic nuances, context-dependencies and latent meaning, due to richness of dynamic variants and figurative use of language (Joshi et al., 2015).

The automatic detection of sarcastic expressions often relies on the contrast between positive and negative sentiment (Riloff et al., 2013). This incongruence can be found on a lexical level with sentiment-bearing words, as in ”*I love being ignored*”. In more complex linguistic settings an action or a situation can be perceived as negative, without revealing any affect-related lexical elements. The intention of the speaker as well as common knowledge or shared experience can be key aspects, as in ”*I love waking up at 5 am*”, which can be sarcastic, but not necessarily. Similarly, verbal irony is referred to as saying the opposite of what is meant and based on sentiment contrast (Grice, 1975), whereas situational irony is seen as describing circumstances with unexpected consequences (Lucariello, 1994; Shelley, 2001).

Empirical studies have shown that there are specific linguistic cues and combinations of such that can serve as indicators for sarcastic and ironic expressions. Lexical and morpho-syntactic cues include exclamations and interjections, typographic markers such as all caps, quotation marks and emoticons, intensifiers and hyperboles (Kunneman et al., 2015; Bharti et al., 2016). In the case of Twitter, the usage of emojis and hashtags has also proven to help automatic irony detection.

We propose a purely character-based architecture which tackles these challenges by allowing us to use a learned representation that models features derived from morpho-syntactic cues. To do so, we use deep contextualized word representations, which have recently been used to achieve the state of the art on six NLP tasks, including sentiment analysis (Peters et al., 2018). We test our proposed architecture on 7 different irony/sarcasm datasets derived from 3 different data sources, providing state-of-the-art performance in 6 of them and otherwise offering competitive results, showing the effectiveness of our proposal. We make our code available at https://github.com/epochx/elmo4irony.

Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 2–7
Brussels, Belgium, October 31, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

2 Related work

Apart from the relevance for industry applications related to sentiment analysis, sarcasm and irony detection has received great traction within the NLP research community, resulting in a variety of methods, shared tasks and benchmark datasets. Computational approaches for the classification task range from rule-based systems (Riloff et al., 2013; Bharti et al., 2015) and statistical methods and machine learning algorithms such as Support Vector Machines (Joshi et al., 2015; Tungthamthiti et al., 2010), Naive Bayes and Decision Trees (Reyes et al., 2013) leveraging extensive feature sets, to deep learning-based approaches. In this context, Tay et al. (2018). delivered state-of-the-art results by using an intra-attentional component in addition to a recurrent neural network. Previous work such as the one by Veale (2016) had proposed a convolutional long-short-term memory network (CNN-LSTM-DNN) that also achieved excellent results. A comprehensive survey on automatic sarcasm detection was done by Joshi et al. (2016), while computational irony detection was reviewed by Wallace (2015).

Further improvements both in terms of classic and deep models came as a result of the SemEval 2018 Shared Task on Irony in English Tweets (Van Hee et al., 2018). The system that achieved the best results was hybrid, namely, a densely-connected BiLSTM with a multi-task learning strategy, which also makes use of features such as POS tags and lexicons (Wu et al., 2018).

3 Proposed Approach

The wide spectrum of linguistic cues that can serve as indicators for sarcastic and ironic expressions has been usually exploited for automatic sarcasm or irony detection by modeling them in the form of binary features in traditional machine learning.

On the other hand, deep models for irony and sarcasm detection, which are currently offer state-of-the-art performance, have exploited sequential neural networks such as LSTMs and GRUs (Veale, 2016; Zhang et al., 2016) on top of distributed word representations. Recently, in addition to using a sequential model, Tay et al. (2018) proposed to use intra-attention to compare elements in a sequence against themselves. This allowed the model to better capture word-to-word level interactions that could also be useful for detecting sarcasm, such as the *incongruity* phenomenon (Joshi et al., 2015). Despite this, all models in the literature rely on word-level representations, which keeps the models from being able to easily capture some of the lexical and morpho-syntactic cues known to denote irony, such as all caps, quotation marks and emoticons, and in Twitter, also emojis and hashtags.

The usage of a purely character-based input would allow us to directly recover and model these features. Consequently, our architecture is based on Embeddings from Language Model or ELMo (Peters et al., 2018). The ELMo layer allows to recover a rich 1,024-dimensional dense vector for each word. Using CNNs, each vector is built upon the characters that compose the underlying words. As ELMo also contains a deep bi-directional LSTM on top of this character-derived vectors, each word-level embedding contains contextual information from their surroundings. Concretely, we use a pre-trained ELMo model, obtained using the 1 Billion Word Benchmark which contains about 800M tokens of news crawl data from WMT 2011 (Chelba et al., 2014).

Subsequently, the contextualized embeddings are passed on to a BiLSTM with 2,048 hidden units. We aggregate the LSTM hidden states using max-pooling, which in our preliminary experiments offered us better results, and feed the resulting vector to a 2-layer feed-forward network, where each layer has 512 units. The output of this is then fed to the final layer of the model, which performs the binary classification.

4 Experimental Setup

We test our proposed approach for binary classification on either sarcasm or irony, on seven benchmark datasets retrieved from different media sources. Below we describe each dataset, please see Table 1 below for a summary.

Twitter: We use the Twitter dataset provided for the SemEval 2018 Task 3, Irony Detection in English Tweets (Van Hee et al., 2018). The dataset was manually annotated using binary labels. We also use the dataset by Riloff et al. (2013), which is manually annotated for sarcasm. Finally, we use the dataset by Ptáček et al. (2014), who collected a user self-annotated corpus of tweets with the *#sarcasm* hashtag.

Reddit: Khodak et al. (2017) collected SARC, a corpus comprising of 600.000 sarcastic comments on Reddit. We use main subset, *SARC 2.0*,

Reference	Dataset	Train	Valid	Test	Total	Source
Van Hee et al., 2018	SemEval-2018	3,067	306	784	3,834	Twitter
Ptáček et al., 2014	Ptáček	48,007	6,858	13,717	68,582	Twitter
Riloff et al., 2013	Riloff	1,327	189	381	1,897	Twitter
Khodak et al., 2017	SARC 2.0	205,665	51,417	64,666	321,748	Reddit
Khodak et al., 2017	SARC 2.0 pol	10,934	2,734	3,406	17,074	Reddit
Oraby et al., 2016	SC-V1	1,396	199	400	1,995	Dialogues
Oraby et al., 2016	SC-V2	3,284	469	939	4,692	Dialogues

Table 1: Benchmark datasets: Tweets, Reddit posts and online debates for sarcasm and irony detection.

and the political subset, *SARC 2.0 pol*.

Online Dialogues: We utilize the *Sarcasm Corpus V1* (SC-V1) and the *Sarcasm Corpus V2* (SC-V2), which are subsets of the Internet Argument Corpus (IAC). Compared to other datasets in our selection, these differ mainly in text length and structure complexity (Oraby et al., 2016).

In Table 1, we see a notable difference in terms of size among the Twitter datasets. Given this circumstance, and in light of the findings by Van Hee et al. (2018), we are interested in studying how the addition of external soft-annotated data impacts on the performance. Thus, in addition to the datasets introduced before, we use two corpora for augmentation purposes. The first dataset was collected using the Twitter API, targeting tweets with the hashtags *#sarcasm* or *#irony*, resulting on a total of 180,000 and 45,000 tweets respectively. On the other hand, to obtain non-sarcastic and non-ironic tweets, we relied on the SemEval 2018 Task 1 dataset (Mohammad et al., 2018). To augment each dataset with our external data, we first filter out tweets that are not in English using language guessing systems. We later extract all the hashtags in each target dataset and proceed to augment only using those external tweets that contain any of these hashtags. This allows us to, for each class, add a total of 36,835 tweets for the Ptáček corpus, 8,095 for the Riloff corpus and 26,168 for the SemEval-2018 corpus.

In terms of pre-processing, as in our case the preservation of morphological structures is crucial, the amount of normalization is minimal. Concretely, we forgo stemming or lemmatizing, punctuation removal and lowercasing. We limit ourselves to replacing user mentions and URLs with one generic token respectively. In the case of the SemEval-2018 dataset, an additional step was to remove the hashtags *#sarcasm*, *#irony* and *#not*, as they are the artifacts used for creating the dataset. For tokenizing, we use a variation of the Twokenizer (Gimpel et al., 2011) to better deal with emojis.

Our models are trained using Adam with a learning rate of 0.001 and a decay rate of 0.5 when there is no improvement on the accuracy on the validation set, which we use to select the best models. We also experimented using a slanted triangular learning rate scheme, which was shown by Howard and Ruder (2018) to deliver excellent results on several tasks, but in practice we did not obtain significant differences. We experimented with batch sizes of 16, 32 and 64, and dropouts ranging from 0.1 to 0.5. The size of the LSTM hidden layer was fixed to 1,024, based on our preliminary experiments. We do not train the ELMo embeddings, but allow their dropouts to be active during training.

5 Results

Table 2 summarizes our results. For each dataset, the top row denotes our baseline and the second row shows our best comparable model. Rows with FULL models denote our best single model trained with all the development available data, without any other preprocessing other than mentioned in the previous section. In the case of the Twitter datasets, rows indicated as AUG refer to our the models trained using the augmented version of the corresponding datasets.

For the case of the SemEval-2018 dataset we use the best performing model from the Shared Task as a baseline, taken from the task description paper (Van Hee et al., 2018). As the winning system is a voting-based ensemble of 10 models, for comparison, we report results using an equivalent setting. For the Riloff, Ptáček, SC-V1 and SC-V2 datasets, our baseline models are taken directly from Tay et al. (2018). As their pre-processing includes truncating sentence lengths at 40 and 80

Dataset		Model	Accuracy	Precision	Recall	F1-Score
Twitter	SemEval-2018	Wu et al. (2018)	**0.735**	0.630	**0.801**	**0.705**
		ELMo-BiLSTM	0.708	**0.696**	0.697	0.696
		ELMo-BiLSTM-FULL	0.702	0.689	0.689	0.689
		ELMo-BiLSTM-AUG	0.658	0.651	0.657	0.651
	Riloff	Tay et al. (2018)	0.823	0.738	0.732	0.732
		ELMo-BiLSTM	0.842	0.759	**0.750**	**0.759**
		ELMo-BiLSTM-FULL	**0.858**	**0.778**	0.735	0.753
		ELMo-BiLSTM-AUG	0.798	0.684	0.708	0.694
	Ptáček	Tay et al. (2018)	0.864	0.861	0.858	0.860
		ELMo-BiLSTM	**0.876**	0.868	0.869	0.869
		ELMo-BiLSTM-FULL	0.872	**0.872**	**0.872**	**0.872**
		ELMo-BiLSTM-AUG	0.859	0.859	0.858	0.859
Dialog	SC-V1	Tay et al. (2018)	0.632	0.639	0.637	0.632
		ELMo-BiLSTM	**0.646**	**0.650**	**0.646**	**0.644**
		ELMo-BiLSTM-FULL	0.633	0.633	0.633	0.633
	SC-V2	Tay et al. (2018)	0.729	0.729	0.729	0.728
		ELMo-BiLSTM	0.748	0.748	0.747	0.747
		ELMo-BiLSTM-FULL	**0.760**	**0.760**	**0.760**	**0.760**
Reddit	SARC 2.0	Khodak et al. (2017)	0.758	-	-	-
		ELMo-BiLSTM	**0.773**	-	-	-
		ELMo-BiLSTM-FULL	0.702	0.760	0.760	0.760
	SARC 2.0 pol	Khodak et al. (2017)	0.765	-	-	-
		ELMo-BiLSTM	**0.785**	-	-	-
		ELMo-BiLSTM-FULL	0.720	0.720	0.720	0.720

Table 2: Summary of our obtained results.

tokens for the Twitter and Dialog datasets respectively, while always removing examples with less than 5 tokens, we replicate those steps and report our results under these settings. Finally, for the Reddit datasets, our baselines are taken from Khodak et al. (2017). Although their models are trained for binary classification, instead of reporting the performance in terms of standard classification evaluation metrics, their proposed evaluation task is predicting which of two given statements that share the same context is sarcastic, with performance measured solely by accuracy. We follow this and report our results.

In summary, we see our introduced models are able to outperform all previously proposed methods for all metrics, except for the SemEval-2018 best system. Although our approach yields higher Precision, it is not able to reach the given Recall and F1-Score. We note that in terms of single-model architectures, our setting offers increased performance compared to Wu et al. (2018) and their obtained F1-score of 0.674. Moreover, our system does so without requiring external features or multi-task learning. For the other tasks we are able to outperform Tay et al. (2018) without requiring any kind of intra-attention. This shows the effectiveness of using pre-trained character-based word representations, that allow us to recover many of the morpho-syntactic cues that tend to denote irony and sarcasm.

Finally, our experiments showed that enlarging existing Twitter datasets by adding external soft-labeled data from the same media source does not yield improvements in the overall performance. This complies with the observations made by Van Hee et al. (2018). Since we have designed our augmentation tactics to maximize the overlap in terms of topic, we believe the soft-annotated nature of the additional data we have used is the reason that keeps the model from improving further.

6 Conclusions

We have presented a deep learning model based on character-level word representations obtained from ELMo. It is able to obtain the state of the art in sarcasm and irony detection in 6 out of 7 datasets derived from 3 different data sources. Our results also showed that the model does not benefit from using additional soft-labeled data in any of the three tested Twitter datasets, showing that manually-annotated data may be needed in order to improve the performance in this way.

References

S. K. Bharti, B. Vachha, R. K. Pradhan, K. S. Babu, and S. K. Jena. 2016. Sarcastic sentiment detection in tweets streamed in real time: a big data approach. *Digital Communications and Networks*, 2(3):108–121.

Santosh Kumar Bharti, Korra Sathya Babu, and Sanjay Kumar Jena. 2015. Parsing-based Sarcasm Sentiment Recognition in Twitter Data. *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15*, pages 1373–1380.

Pranali Chaudhari and Chaitali Chandankhede. 2017. Literature Survey of Sarcasm Detection. pages 2041–2046.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 2635–2639.

Elena Filatova. 2012. Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing. *Lrec*, pages 392–398.

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

H. Paul Grice. 1975. Logic and Conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics*, volume 3, pages 41–58. Academic Press, New York.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339. Association for Computational Linguistics.

Aditya Joshi, Pushpak Bhattacharyya, and Mark James Carman. 2016. Automatic Sarcasm Detection: A Survey. 50(5).

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing Context Incongruity for Sarcasm Detection. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, 51(4):757–762.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A Large Self-Annotated Corpus for Sarcasm.

Florian Kunneman, Christine Liebrecht, Margotvan Mulken, and Antalvan den Bosch. 2015. Signalling sarcasm : From hyperbole to hashtag. *Information Processing & Management*.

Joan Lucariello. 1994. Situational Irony: A Concept of Events Gone Awry. *Journal of Experimental Psychology: General*, 123(2):129–145.

Saif M. Mohammad, Felipe Bravo-Marquez, Salameh Mohammad, and Svetlana Kiritchenko. 2018. SemEval-2018 Task 1 : Affect in Tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 1–17.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue. *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, (September):31–41.

Bo Pang and Lillian Lee. 2006. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 1(2):91–231.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Tomáš Ptáček, Ivan Habernal, and Jun Hong. 2014. Sarcasm Detection on Czech and English Twitter. *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING 2014), Citeseer*, pages 213–223.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in Twitter. *Language Resources and Evaluation*, 47(1):239–268.

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. *Emnlp*, (Emnlp):704–714.

Cameron Shelley. 2001. The bicoherence theory of situational irony. *Cognitive Science*, 25(5):775–818.

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

Piyoros Tungthamthiti, Kiyoaki Shirai, and Masnizah Mohd. 2010. Recognition of Sarcasm in Microblogging Based on Sentiment Analysis and Coherence

Identification. *Journal of Natural Language Processing*, 23(5):383–405.

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 39–50.

Tony Veale. 2016. Fracking Sarcasm using Neural Network Fracking Sarcasm using Neural Network. *Acl*, (May):161–169.

Byron C. Wallace. 2015. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, 43(4):467–483.

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 2018. THU NGN at SemEval-2018 Task 3 : Tweet Irony Detection with Densely THU NGN at SemEval-2018 Task 3 : Tweet Irony Detection with Densely Connected LSTM and Multi-task Learning. (March):51–56.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460. The COLING 2016 Organizing Committee.