Masking and Generation: An Unsupervised Method for Sarcasm Detection

Rui Wang
Joint Lab of CMS-HITSZ,
Harbin Institute of Technology
Shenzhen, China
ruiwangnlp@outlook.com

Qianlong Wang
Joint Lab of CMS-HITSZ,
Harbin Institute of Technology
Shenzhen, China
qlwang15@outlook.com

Bin Liang
Joint Lab of CMS-HITSZ,
Harbin Institute of Technology
Shenzhen, China
bin.liang@stu.hit.edu.cn

Yi Chen
Joint Lab of CMS-HITSZ,
Harbin Institute of Technology
Shenzhen, China
yichennlp@gmail.com

Zhiyuan Wen
Joint Lab of CMS-HITSZ,
Harbin Institute of Technology
Shenzhen, China
wenzhiyuan2012@gmail.com

Bing Qin
Harbin Institute of Technology
Harbin, China
qinb@ir.hit.edu.cn

Ruifeng Xu*
Harbin Institute of Technology
Shenzhen, China
Peng Cheng Laboratory
Shenzhen, China
xuruifeng@hit.edu.cn

ABSTRACT

Existing approaches for sarcasm detection are mainly based on supervised learning, in which the promising performance largely depends on a considerable amount of labeled data or extra information. In the real world scenario, however, the abundant labeled data or extra information requires high labor cost, not to mention that sufficient annotated data is unavailable in many low-resource conditions. To alleviate this dilemma, we investigate sarcasm detection from an unsupervised perspective, in which we explore a masking and generation paradigm in the context to extract the context incongruities for learning sarcastic expression. Further, to improve the feature representations of the sentences, we use unsupervised contrastive learning to improve the sentence representation based on the standard dropout. Experimental results on six perceived sarcasm detection benchmark datasets show that our approach outperforms baselines. Simultaneously, our unsupervised method obtains comparative performance with supervised methods for the intended sarcasm dataset.

CCS CONCEPTS

• **Information systems → Sentiment analysis**; **Information extraction**; **Clustering and classification**; *Information retrieval.*

KEYWORDS

Words mask, Natural language generation, Unsupervised sarcasm detection, Pre-trained language models, Sentiment analysis

ACM Reference Format:
Rui Wang, Qianlong Wang, Bin Liang, Yi Chen, Zhiyuan Wen, Bing Qin, and Ruifeng Xu. 2022. Masking and Generation: An Unsupervised Method for Sarcasm Detection. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477495.3531825

1 INTRODUCTION

Sarcasm is a sophisticated language phenomenon, which would cause much confusion to exist sentiment classification systems. So sarcasm detection, a task of predicting whether a given text contains sarcasm, has received much research attention [1–3, 12, 13, 17, 18, 22, 23]. As shown in Figure 1, we only need to detect the first sentence as sarcastic. Although both sentences contain a decisive sentiment word "*love*", the word "*ignored*" leads the whole sentence to an opposite sentiment polarity. That is the incongruity expressions of sarcasm context.

Recently, many methods have been proposed for sarcasm detection, which could be broadly classified into two categories. One is the text-only method which only concentrate on the utterance itself , such as exploiting incongruity expressions to detect the sarcasm text [9, 23]. Another direction is based on extra information, which exploits external knowledge to assist the detection procedure, such as user history [19, 21], and common sense knowledge [26].

Although these two kinds of methods have achieved satisfying performance, the procedure of collecting annotated data or other extra knowledge is tedious work. Meanwhile, a sarcastic utterance might not be perceived because of the different backgrounds of

[*] Corresponding Author.

SIGIR ’22, July 11–15, 2022, Madrid, Spain.
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531825

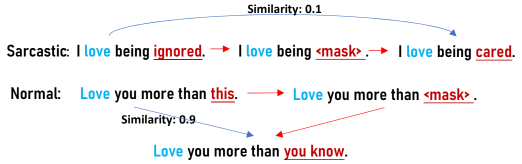
Similarity: 0.1
Sarcastic: I love being ignored. → I love being <mask>. → I love being cared.
Normal: Love you more than this. → Love you more than <mask>.
Similarity: 0.9
Love you more than you know.

Figure 1: Red denotes the masked places and blue means decisive sentiment words. During the mask and generation procedure, sarcastic texts suffer more changes than normal texts. Hence, for sarcasm sentences, the similarity between original and reborn texts will be relatively lower.

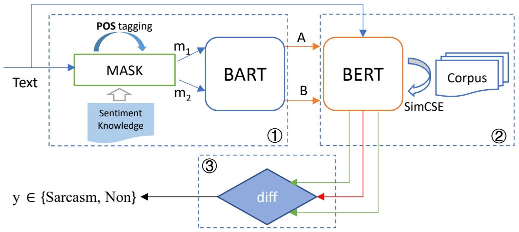

Figure 2: Architecture of our proposed method.

audiences [24], which will lead to serious annotation errors. To alleviate these problems, Joshi et al. [11] proposed an unsupervised method, which compares the word observed in the original sentence and the one expected after the sentence completion. However, they only considered a single token to reconstruct sentences rather than phrases. This is insufficient because most of the sarcastic incongruity is composed of phrases. Consequently, Joshi et al. acquired higher performance in short text datasets since this method poorly captures the incongruity of longer sarcastic texts.

Sarcastic expressions tend to possess more sentiment inconsistencies and logical conflicts. If we properly mask some words in the sentence, the sentiment and logical contradictions will be influenced. We then can obtain a new sentence by feeding the remaining parts of the sentence into a pre-trained generation model. Since the pre-trained generation model is pre-trained on general corpora where sarcastic texts are scarce, we assume that given a masked text, it can generate a relatively normal one according to the remaining logic information. As the example shown in Figure 1, the sentence with the masked word "*ignored*" is regenerated into a normal text.

Based on these assumptions, we propose an unsupervised sarcasm detection method. First, we leverage the external sentiment knowledge and POS information to mask prominent tokens. Then the masked texts are fed into the pre-trained generation model, which follows the remaining logic structure to generate texts. There is a good chance that these reborn texts would not be sarcastic or make more sense. Second, after obtaining the similarity score between the generated sentence and the original one, features beneath the scores will be extracted to decide whether a sentence is sarcasm. In addition, to overcome the anisotropy problem [7, 15], we employ unsupervised contrastive learning [8] to obtain a better sentence representation, which contributes to improving system performance.

We construct several unsupervised baselines and conduct experiments on seven datasets, including perceived sarcasm datasets and intended sarcasm dataset. Our method obtains comparative or better performance than baseline models. Especially for the intended sarcasm dataset, our method surpasses many supervised approaches.

2 MODEL

2.1 Problem Formulation

The sarcasm detection task can be characterized as a binary classification task. Given a token sequence $x = \{x_1, x_2, ..., x_n\}$ of length n, the sarcasm detection model takes x as input and outputs a label $y \in \{Sarcasm, Non\}$.

2.2 Overview

Figure 2 provides an illustration of our method. The proposed framework contains three main components: 1) *Sentences mask and generation.* This procedure first recognizes main components of sentences which will be properly masked to cause more impact on original sentences, and then fulfills the texts generation work; 2) *Sentences representation.* It is expected to calculate dense vectors of sentences, and further is finetuned by unsupervised contrastive learning to obtain better representation; 3) *Sarcastic utterances detection* leverages the similarity scores between original and regenerated sentences to detect whether an utterance is sarcastic.

2.3 Sentences Mask and Generation

First, we use the sentiment common knowledge retrieved from SenticNet [5] to recognize affective words in the sentence x, and split those words into two sets according to its sentiment polarities:

$$PW = \{pw_1, pw_2, ..., pw_h\}$$
$$NW = \{nw_1, nw_2, ..., nw_k\}, h + k \leq n$$

Second, we analyze the sentence to get its syntax information[1] to identify non-stop words $SW = \{sw_1, sw_2, ..., sw_m, m \leq n\}$. Intuitively, these words are the main components of sentences. Then we split SW into two sets which satisfy :

$$SW_1 \cup SW_2 = SW, \ |SW_1| = |SW_2|$$

Here, $PW \cup SW_1$ and $NW \cup SW_2$ are used to mask original sentence respectively. So we will obtain two masked sentences $x_{m1} = \{[m]_1, x_2, ..., [m]_n\}$ and $x_{m2} = \{x_1, [m]_2, ..., x_n\}$. These two masked sentences are fed into the pre-trained generation model to fulfill the generation procedure.

$$\boldsymbol{A}\{a_1, ..., x_2, ..., x_{n-1}, ..., a_o\} = BART([m]_1, x_2, ..., x_{n-1}, [m]_n) \quad (1)$$

Thus, we will obtain two reborn sentences $A = \{a_1, a_2, ..., a_o\}$ and $B = \{b_1, b_2, ..., b_p\}$. Since the pre-trained language model is pre-trained on general corpora, it is prone to generate an affective fluency sentence rather than a sarcastic one. Consequently, this generation procedure will cause more impact on sarcastic texts than normal texts which is the basic assumption of our method.

2.4 Sentences Representation

We embed the original sentence x and its corresponding reborn texts A and B into d-dimentional embedding $\boldsymbol{H_t} \in \mathbb{R}^d$ via pre-trained

[1] We employ SpaCy to finish POS tagging: https://spacy.io/ .

Table 1: Statistics of training and test datasets.

Dataset	Train		Test	
	Sarcasm	Non	Sarcasm	Non
IAC-V1	862	859	97	94
IAC-V2	2, 947	2, 921	313	339
Tweet-1	23, 456	24, 387	2, 569	2, 634
Tweet-2	282	1, 051	35	113
Reddit-1	5, 521	5, 607	1, 389	1, 393
Reddit-2	6, 419	6, 393	1, 596	1, 607
iSarcasm	476	2, 346	124	582

BERT-base [6]:

$$\boldsymbol{H}_x, \boldsymbol{H}_A, \boldsymbol{H}_B = BERT(x), BERT(A), BERT(B) \tag{2}$$

Since sentences vectors of pre-trained models are not uniformly distributed with respect to direction, we can employ SimCSE [8] on datasets to further improve BERT's embedding.

2.5 Sarcastic Utterances Detection

We utilize cosine similarity to measure the similarity between representations of original sentence H_x and generation texts H_A/H_B. Then we use the following equation to calculate a *difference score* of each sentence:

$$\mathbf{diff} = \mathbf{sim}(H_x, H_A) < threshold \;||\; \mathbf{sim}(H_x, H_B) < threshold \tag{3}$$

where || means "*or*" logical operator. Since the sarcastic utterances are influenced more than normal texts during the masking and generation procedure, the *difference score* of sarcastic texts should be greater than a non-sarcastic one. If we have a *threshold* value which separates sarcastic texts and normal texts, we can yield the prediction y by:

$$y = \mathbb{I}(\mathbf{diff}) \tag{4}$$

3 EXPERIMENTS

3.1 Experimental Data and Settings

Datasets. We conduct experiments on seven Sarcasm datasets: **six** Perceived Sarcasm datasets from **Twitter** [22, 23], **Reddit** [12], and **IAC** [18]; **one** Intended Sarcasm dataset (**iSarcasm**) [20]. For Twitter dataset, we retrieve tweets using Twitter API with the provided tweet IDs[2]. Statistics of the datasets are presented in Table 1. Note that the proposed unsupervised method uses only the text in the train set.

Settings. We utilize the pre-trained BART-base [14] as the text generation module and uncased BERT-base [6] as the sentence representation module. For the contrastive learning procedure, the learning rate is $3e - 5$, the mini-batch size is 32 for all datasets, and τ is 0.05. Our method is dependent on the threshold value. Hence, we use 20% labeled data in the train set as the development set to select the optimal threshold, and report results on the test set. We employ the F1 metric to evaluate the performance of the models.

3.2 Comparison Baselines

We compare our model with four baselines. 1) **Lexicon**. We use SenticNet to identify the positive and negative words in the sentence. If a sentence contains positive and negative words at the same time, we regard the sentence as sarcastic. 2) **TF-IDF-LDA** and 3) **TF-IDF-Kmeans**. Here, TF-IDF coefficients are utilized as sentence representations. The data is clustered into two clusters by Latent Dirichlet Allocation [4] and K-Means [16] algorithm respectively. 4) **BERT with Words Mask** [11]. We re-implement this unsupervised model and replace its sentence completion module with BERT. We also replicate supervised results on the *iSarcasm* dataset's reported by Opera et al. [20] to compare with our method.

3.3 Main Experiment Results

Table 3 shows the experiment results on six benchmark datasets. We can draw the following conclusion: 1) We can observe that our proposed method outperforms all compared baselines on all datasets. Specifically, the best improved results of Acc. and F1 respectively are 9.69% and 11.83% compared with baselines. 2) Our proposed paradigm far exceeds BERT+word-Mask, suggesting that only considering *one* word can not fully represent sarcastic meanings. 3) The employment of SimCSE in our model further improves performance, which means unsupervised contrastive learning contributes to better sentences representation.

Furthermore, as shown in Table 4, our method surpasses unsupervised baselines and previous supervised sarcasm detection approaches on the intended dataset. For example, compared with MIARN which has been reported the best scores, the proposed method improves 15.65% F1 score. This verifies that our proposed masking and generation paradigm significantly captures the text incongruities and effectively improves the performance of sarcasm detection. Surprisingly, unsupervised methods **Lexicon** and **TF-IDF-LDA** outperform supervised methods. The underlying reason is that the sarcasm in the iSarcasm dataset is lexically shallow and semantically deep.

Table 2: Ablation study (F1). $\mathcal{S}$ denotes the split of affective words. $\mathcal{M}$+$\mathcal{G}$ denotes the Masking and Generation procedure.

MODEL	IAC1	IAC2	Tweet-1	Tweet-2	Reddit-1	Reddit-2
Our	55.44	63.17	58.76	58.31	54.91	55.80
w/o $\mathcal{S}$	54.32	60.26	56.23	57.75	52.52	54.58
w/o $\mathcal{M} + \mathcal{G}$	33.68	54.22	33.36	52.53	43.29	53.09

3.4 Ablation Study

To analyze the impact of different components of our proposed Masking and Generation method, we conduct an ablation study in Table 2. 1) *w/o* $\mathcal{S}$ variant mixes the positive and negative words rather than split them into two different groups. The mixture of affective words leads to considerably poorer performance. 2) *w/o* $\mathcal{M} + \mathcal{G}$ variant uses just representations from SimCSE-BERT to finish clustering by LDA and K-Means. We report the best F1 results. We see that the removal of mask and generation procedure sharply degrades performance, which indicates that our proposed paradigm plays a significant role in the understanding of the sarcastic expression. 3) Table 3 and Table 4 show that contrastive learning indeed improves representation quality of BERT. The performance would drop dramatically without SimCSE.

4 DISCUSSION

Impact of Different Prediction Ways on Performance. In the previous experiment, we use the **rule-1** (Eq. 3 and Eq. 4) to

[2] http://api.twitter.com/

Table 3: Main experimental results on different datasets. Average scores over five runs are reported. Best scores are in bold. Second best scores are underlined.

MODEL	IAC1		IAC2		Tweet-1		Tweet-2		Reddit-1		Reddit-2	
	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)	Acc.(%)	F1.(%)
Lexicon	47.64	40.29	44.01	39.12	<u>59.00</u>	55.86	57.43	51.7	43.06	42.71	42.77	41.47
TF-IDF-LDA	<u>53.40</u>	53.22	54.61	52.44	54.52	54.36	50.68	48.15	52.51	50.81	51.72	47.65
TF-IDF-Kmeans	49.73	49.35	51.68	47.52	52.27	44.1	**72.97**	51.86	49.68	46.74	<u>52.58</u>	43.29
BERT+word-Mask [11]	51.39	36.35	48.00	35.72	**59.46**	<u>56.54</u>	41.22	41.21	47.19	39.47	46.91	37.63
Ours	52.35	<u>53.75</u>	<u>62.06</u>	<u>56.75</u>	50.21	52.35	<u>67.57</u>	<u>55.24</u>	<u>52.62</u>	<u>52.60</u>	51.92	<u>49.89</u>
Ours+SimCSE	**57.59**	**55.44**	**64.30**	**64.27**	58.91	**58.76**	56.76	**58.31**	**53.30**	**54.91**	**56.16**	**56.14**

Table 4: Experiment results on iSarcasm Dataset. Best Scores are in bold. Second best scores are underlined.

MODEL	Precision.(%)	Recall.(%)	F1.(%)
Lexicon	49.2	48.7	40.5
TF-IDF-LDA	15.7	49.0	42.6
TF-IDF-Kmeans	18.8	32.5	32.4
BERT+word-Mask [11]	16.7	**88.5**	24.0
LSTM	21.7	74.7	33.6
CNN	26.1	56.3	35.6
SIARN [25]	21.9	78.2	34.2
MIARN [25]	23.6	<u>79.3</u>	36.4
3CNN [10]	25.0	33.3	28.6
Dense-LSTM [27]	<u>37.5</u>	27.6	31.8
Ours	**50.7**	50.5	<u>50.1</u>
Ours+SimCSE	20.5	72.7	**52.1**

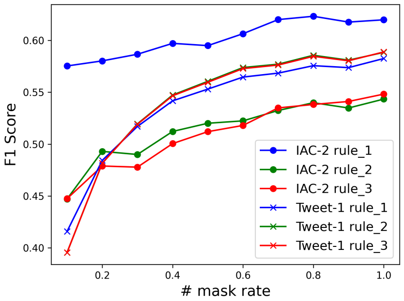

Figure 3: The performance of different prediction rules and mask rates. The average lengths of texts are <u>270</u> and <u>80</u> for IAC-2 and Tweet-1 respectively.

Table 5: Different numbers of labeled data (development set) and corresponding thresholds. Average scores over 100 runs are reported. *T*- and *R*- represent Tweet and Reddit respectively.

number	10	50	200	10%	best-F1
T-threshold	0.9920	0.9839	0.9759	0.9759	0.9820
R-threshold	0.9076	0.9437	0.9317	0.9317	0.9260
Tweet-1	52.91	55.85	56.67	57.08	58.76
Reddit-2	50.35	53.76	55.02	55.03	56.14

finish sarcasm detection. Here, we explore the effect of another two prediction rules:

$$\begin{aligned} \textbf{rule-2}: \quad & y = \mathbb{I}(\mid \textbf{sim}(H_x, H_A) - \textbf{sim}(H_x, H_B) \mid < \textit{threshold}) \\ \textbf{rule-3}: \quad & y = \mathbb{I}(\textbf{sim}(H_A, H_B) < \textit{threshold}) \end{aligned} \tag{5}$$

The experiment results are reported in Figure 3. The curves of Tweet-1 gather together which means it is almost not influenced by different rules. For longer texts such as IAC-2, **rule-1** obtaines the best performance.

Impact of different Masking Rates on Performance. In the previous experiment, we mask all affective words and other non-stop tokens to yield two incomplete sentences. Here, we change the mask rate and report performance. From Figure 3, we can find that a higher mask rate can improve performance. Besides, the datasets (e.g., IAC-2) which have more long texts tend to be less affected by changes of mask rates.

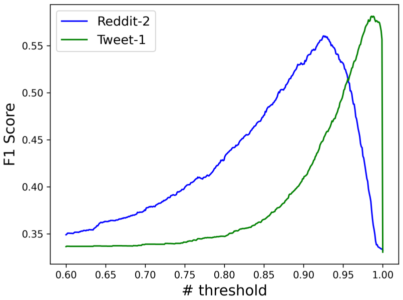

Figure 4: Thresholds and corresponding F1 scores on two datasets.

Impact of Different Thresholds on Performance. To explore the impact of *threshold* values, we vary thresholds and report results in Figure 4. The curves are sharp which means that small perturbation to the best threshold will degrade performance. In addition, as shown in Table 5, we argue that a few labeled texts could effectively approach the best threshold and obtain a promising performance.

Case Study. We present the predictions of models on three random examples in Figure 5. We can conclude that our model is capable to handle more complex semantic and affective information and longer texts. 1) Sentence 1 is a simple example that clearly contains positive and negative sentiment words. The word *"stupid"* and *"awesome"* form a sentiment contradict which can be easily detected by **Lexicon** and other methods. 2) Sentence 2 only appears positive sentiment words. Therefore, it is difficult to detect sarcasm just by affective information. However, after word mask and prediction via BERT, we can obtain a dissimilar comparison between the original word *"love"* and the retrieved word *"keep"*. 3) Sentence 3 is a more complex example, since its sentiment and logical contradictions are

	Lexicon	BERT+word-mask	Ours
1. yes cause a stupid looking Duck on a hat is pretty awesome. *Sarcastic*	√	√	√
2. I just love getting calls from restricted numbers. *Sarcastic* -> I just keep getting calls from restricted numbers. Retrieved word by BERT+word-mask	×	√	√
3. So the Romans nailed anyone up that organized the community! Did you get that from the film? *Sarcastic* -> So the Romans nailed anyone who did that to the community! Did you get that from the film? Generated text by Ours -> So the Romans didn't set anyone up that organized the event. Did you know that from the book? Generated text by Ours	×	×	√

Figure 5: Three examples for case study. Red denotes negative word. Blue denotes positive word. Purple represents masked tokens. Green represents corresponding generated tokens.

deduced from phrases rather than words. **BERT+word-mask** is not qualified for processing this case. However, through masking and generation procedure, the pre-trained language model infers common sense knowledge from the remaining logical structure of the text and generates other sentences. Hence, we can decide that Sentence 3 is sarcastic by comparison between several generated sentences.

5 CONCLUSION

In this paper, we propose an unsupervised sarcasm detection method, which reveals the context incongruities in the sarcastic texts via the masking and generation paradigm. Specifically, we first identify the common sentiment knowledge and POS information, in which some words will be appropriately masked. Then, we feed the masked sentences into the pre-trained models to repair sentences. Finally, we perform the sarcasm detection task based on the similarity between original and reborn texts. Experimental results show that our approach outperforms baselines on six perceived sarcasm datasets, and obtains comparative performance with supervised methods on the intended sarcasm dataset.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (61876053, 62006062, 62176076), the Shenzhen Foundational Research Funding JCYJ20210324115614039, Shenzhen Science and Technology Program JSGG20210802154400001, and the Joint Lab of HITSZ and China Merchants Securities.

REFERENCES

[1] Ameeta Agrawal and Aijun An. 2018. Affective Representations for Sarcasm Detection. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 1029–1032. https://doi.org/10.1145/3209978.3210148

[2] Ameeta Agrawal, Aijun An, and Manos Papagelis. 2020. *Leveraging Transitions of Emotions for Sarcasm Detection.* Association for Computing Machinery, New York, NY, USA, 1505–1508. https://doi.org/10.1145/3397271.3401183

[3] David Bamman and Noah Smith. 2015. Contextualized Sarcasm Detection on Twitter. *Proceedings of the International AAAI Conference on Web and Social Media* 9, 1 (2015), 574–577. https://ojs.aaai.org/index.php/ICWSM/article/view/14655 Number: 1.

[4] David Blei, Andrew Ng, and Michael Jordan. 2001. Latent Dirichlet Allocation. In *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani (Eds.), Vol. 14. MIT Press. https://proceedings.neurips.cc/paper/2001/file/296472c9542ad4d4788d543508116cbc-Paper.pdf

[5] Erik Cambria, Yang Li, Frank Z. Xing, Soujanya Poria, and Kenneth Kwok. 2020. SenticNet 6: Ensemble Application of Symbolic and Subsymbolic AI for Sentiment Analysis. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 105–114. https://doi.org/10.1145/3340531.3412003

[6] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[7] Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 55–65. https://doi.org/10.18653/v1/D19-1006

[8] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 6894–6910. https://doi.org/10.18653/v1/2021.emnlp-main.552

[9] 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*. Association for Computational Linguistics, Portland, Oregon, USA, 581–586. https://aclanthology.org/P11-2102

[10] Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. CASCADE: Contextual Sarcasm Detection in Online Discussion Forums. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 1837–1848. https://aclanthology.org/C18-1156

[11] Aditya Joshi, Samarth Agrawal, Pushpak Bhattacharyya, and Mark J. Carman. 2018. Expect the Unexpected: Harnessing Sentence Completion for Sarcasm Detection. In *Computational Linguistics: 15th PACLING 2017 - Revised Selected Papers*, Vol. 781. 275–287.

[12] Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A Large Self-Annotated Corpus for Sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan. https://aclanthology.org/L18-1102

[13] Amit Kumar Jena, Aman Sinha, and Rohit Agarwal. 2020. C-Net: Contextual Network for Sarcasm Detection. In *Proceedings of the Second Workshop on Figurative Language Processing*. Association for Computational Linguistics, Online, 61–66. https://doi.org/10.18653/v1/2020.figlang-1.8

[14] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[15] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 9119–9130. https://doi.org/10.18653/v1/2020.emnlp-main.733

[16] S. Lloyd. 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 2 (1982), 129–137. https://doi.org/10.1109/TIT.1982.1056489

[17] Chenwei Lou, Bin Liang, Lin Gui, Yulan He, Yixue Dang, and Ruifeng Xu. 2021. Affective Dependency Graph for Sarcasm Detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Virtual Event Canada, 1844–1849. https://doi.org/10.1145/3404835.3463061

[18] Stephanie Lukin and Marilyn Walker. 2013. Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media*. Association for Computational Linguistics, Atlanta, Georgia, 30–40. https://aclanthology.org/W13-1104

[19] 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. Association for Computational Linguistics, Florence, Italy, 2854–2859. https://doi.org/10.18653/v1/P19-1275

[20] 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, Online, 1279–1289. https://doi.org/10.18653/v1/2020.acl-main.118

[21] Joan Plepi and Lucie Flek. 2021. Perceived and Intended Sarcasm Detection with Graph Attention Networks. In *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 4746–4753. https://aclanthology.org/2021.findings-emnlp.408

[22] 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*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, 213–223. https://aclanthology.org/C14-1022

[23] 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*. Association for Computational Linguistics, Seattle, Washington, USA, 704–714. https://aclanthology.org/D13-1066

[24] Patricia Rockwell and Evelyn M. Theriot. 2001. Culture, gender, and gender mix in encoders of sarcasm: A self-assessment analysis. *Communication Research Reports* 18, 1 (2001), 44–52. https://doi.org/10.1080/08824090109384781 arXiv:https://doi.org/10.1080/08824090109384781

[25] 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)*. Association for Computational Linguistics, Melbourne, Australia, 1010–1020. https://doi.org/10.18653/v1/P18-1093

[26] Byron C. Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. 2014. Humans Require Context to Infer Ironic Intent (so Computers Probably do, too). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 512–516. https://doi.org/10.3115/v1/P14-2084

[27] 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 connected LSTM and Multi-task Learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, New Orleans, Louisiana, 51–56. https://doi.org/10.18653/v1/S18-1006