Lexicon-Based Profiling of Irony and Stereotype Spreaders

Notebook for PAN at CLEF 2022

Hyewon Jang[1]

[1]University of Konstanz, Germany

Abstract

The PAN 22 Author Profiling Shared Task (IROSTEREO) aims to profile authors spreading irony and stereotypes on Twitter. In this paper, we report that from our experiments involving different classification methods – traditional n-gram approach, state-of-the-art language models, and lexical approach using LIWC, the best result was obtained from the lexicon-based approach (LIWC) with the accuracy score of 0.88 on the validation data and 0.92 on the official test data. Furthermore, we perform ablation experiments to identify some of the most informative features for irony and stereotype profiling.

Keywords

irony, LIWC, hate speeech, sexism, PAN 22'

1. Introduction

The goal of the PAN '22 Author Profiling Shared Task is to classify authors on social media (Twitter) that disperse stereotypes by using irony, especially towards women and the LGBT community [1, 2]. The task is unique in that it combines two of the commonly addressed tasks in the NLP community: hate speech (and sexism) detection and irony detection.

In this paper, we develop models for author profiling (defined as a binary classification task: "non-ironic" vs. "ironic") by using a variety of features and models. The models with the highest validation score were obtained by employing a lexicon-based feature extractor on traditional classifiers (Support Vector Machine and Random Forest Classifier), which beat the performance of state-of-the-art models. We also perform feature analyses to identify some of the most informative features for the author profiling task.

The organization of this paper is as follows. Section 2 summarizes some of the previous efforts in related tasks. Section 3 describes the proposed methodologies for the PAN '22 Author Profiling Shared Task. Section 4 describes the classification results and Section 5 provides feature analysis results.

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

hye-won.jang@uni-konstanz.de (H. Jang)

2. Related Work

2.1. Irony Detection

(Verbal) Irony is often defined in the literature as a communicative act of expressing the opposite of literal meaning [3, 4, 5]. Although irony is not always associated with hostility, it certainly can be aggressive depending on the context [4, 5]. In computational research, the type of task that is more common than irony detection is sarcasm detection. Sarcasm, which is often considered to be a subset of irony [6], is generally regarded as a more aggressive type of irony [7]. There have been many attempts to detect sarcasm computationally, using methods ranging from traditional classifiers [8, 9, 10] using extracted features [10, 11, 12] to state-of-the-art models [13, 14, 15, 16, 17].

2.2. Hate Speech (and Sexism) Detection

Hate speech is defined by the United Nations as "any kind of communication in speech, writing or behaviour, that attacks or uses pejorative or discriminatory language with reference to a person or a group on the basis of who they are, in other words, based on their religion, ethnicity, nationality, race, colour, descent, gender or other identity factor" [1]. By definition, hate speech detection often gets closely intertwined with sexism detection. A lot of effort has been made to detect or classify hate speech or sexism in online spheres. Some have focused on identifying features useful for hate speech detection: Davidson et al. [18] developed a hate speech lexicon and Waseem and Hovy [19] identified features that improve hate speech classification results. Samory et al. [20] developed a sexism detection dataset using crowdsourcing and adversarial examples. Others have used traditional or state-of-the-art models to detect or classify hate speech: Ceron and Casula [21] used BERT-based architectures to detect hate speech and Parikh et al. [22] used LSTM-based architectures to classify different types of sexism. Anusha and Shashirekha [23] and Zimmerman et al. [24] proposed ensemble methods to classify text into hate speech or non-hate speech.

3. Proposed Methods

3.1. Data and Text Preprocessing

The training dataset for PAN '22 Author Profiling task was provided by the PAN 2022 shared task organizers [25]. The dataset consists of 200 tweets from 420 users, all in English. All the tweets of each user are contained in one xml file and each user is assigned a binary label of whether they are spreading irony and stereotypes ("NI": non-ironic, "I": ironic). Usernames and mentions of particular users in the body of tweets are anonymized before the dataset is shared with the task participants.

[1] https://www.un.org/en/hate-speech/understanding-hate-speech/what-is-hate-speech?gclid=CjwKCAjwyryUBhBSEiwAGN5OCIjIZ8upWu5RHqQEwblwUPUlCAJhLwOsCSjquy48bpLTbFAgGEaaXRoCrYcQAvD_BwE

We extracted and grouped all tweets by each user and their ground-truth labels. Preprocessing was done minimally: mentions of users, hashtags, and url tags were removed. The resulting dataset consisted of tweets from 420 users, with an average word count of 4,338 per user.

3.2. Language Representation Methods

3.2.1. Traditional N-gram Features

As a baseline, we experiment with Term Frequency-Inverse Document Frequency features. After converting the data into a TF-IDF matrix using different N-gram sizes (N=1, 2), unigrams (N=1) proved to be the best in terms of classification performance and computational efficiency.

3.2.2. Sentence-Transformers Embeddings

As many state-of-the-art deep learning-based models have performed well in hate-speech (or sarcasm) detection and irony (sarcasm) detection [13, 14, 15, 26, 27, 21, 22, 24], we experimented with some of the deep learning-based models. We represented the text using the pre-trained embeddings available on the Sentence Transformer library [28]. The pre-trained models used in our experiments were *all-mpnet-base-v2* and *average_word_embeddings_glove.840B.300d.* The former (*all-mpnet-base-v2*) was chosen because it was reported to produce the best quality of sentence embeddings among the general-purpose models [28]. The latter (*average_word_embeddings_glove.840B.300d*), which provides average embeddings from GloVe [29] pre-trained embeddings, was selected because the computation speed is higher than other transformers-based models [28].

3.2.3. Lexicon-based Features

In order to obtain results with better explainability, we also developed a model using only lexicon-based features. We used the software Lexical Inquiry and Word Count (LIWC) (version 2015) to extract features belonging to various lexical categories such as *Psychological Processes* and *Linguistic Dimensions* [30]. This text analysis application extracts the percentage of words belonging to certain lexical categories based on its internal dictionary. A total of 93 features were extracted [2]. To avoid issues arising from range differences between features, we scaled the features by subtracting the mean (u) and scaling them to unit variance by dividing them by the standard deviation (s) as in the following formula. We used the StandardScaler function from the *sklearn* library on Python for this.

```
z = (x - u) / s
```

The scaled features were used as input to the classifiers, which are described in Section 3.3.

[2] See https://mcrc.journalism.wisc.edu/files/2018/04/Manual_LIWC.pdf for all features.

3.3. Classifiers

3.3.1. Traditional Classifiers

Using the features mentioned in 3.2 as input, we experimented with several traditional classifiers: Random Forest Classifier (RF), Support Vector Classifier (SVC), Gaussian Process Classifier (GPC), Decision Tree Classifier (DT), Adaptive Boost Classifier (ABC). We only report results by SVC and RF, which produced the best results.

3.3.2. Transformers-Fine-Tuning

Instead of using contextualized embeddings extracted from transformer-based pre-trained models as input to subsequent classifiers, we also fine-tuned two transformer-based pre-trained models on our dataset: BERT [31] and distilBERT [32]. BERT was chosen because a myriad of research has proven its good performance on a range of NLP tasks [31, 33, 34, 35, 36, 37] DistilBERT was chosen as it is reported to be a more efficient model compared to BERT [32].

3.4. Implementation Details

The original training dataset was split into the training and validation set with a 80 to 20 ratio after the data points were shuffled. With the traditional classifiers, the classification was implemented 100 times with random split and the average accuracy scores are reported as the final performance score to prevent biases stemming from the data split structure.
For the fine-tuning approach, the text by each user was truncated to the maximum number of allowed tokens (512 tokens). The training was carried out for 3 epochs with the batch size of 64. The official metric used for the PAN '22 Author Profiling Task is accuracy score, which is the ratio of the sum of true positives and true negatives out of all the predictions. The evaluation of the models on the official test set was done on the Tira[3] platform [38]. The implementation for the classification experiments was done using the *sklearn*, *torch*, and *transformers* libraries on Python.

4. Results

The full results on the validation data are presented in Table 1. Our best performances come from LIWC-SVC and LIWC-RF (described in 3.2.3 and 3.3.1), with a validation accuracy score of 0.88. We made a submission using the LIWC-SVC model, and the official test set accuracy for the model is 0.92. Our submission ranked the 45th place out of a total of 65 submissions. The submission with the highest accuracy score had the score of 0.99 and the submission with the lowest accuracy score had 0.53.
Our winning models beat the transformers-based fined-tuned models. One possible reason behind this could be the length of each tweet being way over the limit of the maximum word allowed in the transformer models. Both BERT-base and DistilBERT allow up to 512 tokens for

[3]https://www.tira.io/
[33]Official test set accuracy: 0.92

Table 1
Accuracy scores by all models on validation data. ST = Sentence-Transformer embeddings. SVC = Support Vector Classifier. RF = Random Forest Classifier.

Model	Validation Accuracy
TFIDF-SVC	0.87
TFIDF-RF	0.85
ST (average_word_embeddings_glove.840B.300d)-SVC	0.79
ST (average_word_embeddings_glove.840B.300d)-RF	0.85
ST (all-mpnet-base-v2)-SVC	0.78
ST (all-mpnet-base-v2)-RF	0.74
LIWC-SVC	**0.88**[3]
LIWC-RF	**0.88**
DistilBERT-fine-tuned	0.73
BERT-fine-tuned	0.65

Table 2
Accuracy scores (SVC) from the ablation experiment on LIWC features. Top N features ($N \in (10, 20, ...90)$) based on the R-squared values from a simple linear regression model (LM). LM was run between each dimension of LIWC features and the target labels. Boldface indicates the point when the classification performance starts to degrade.

Model	Validation Accuracy
Top 10	**0.76**
Top 20	**0.80**
Top 30	0.88
Top 40	0.87
Top 50	0.88
Top 60	0.88
Top 70	0.88
Top 80	0.88
Top 90	0.88

encoding text. Considering that the mean word count for the tweets is 4,338, a significant amount of information could have been lost in the process of text representation.

5. Post-hoc Feature Analyses

Since LIWC features yielded the highest classification scores, we performed further analyses to probe for any improvement or drop in the classification performance by subtracting certain features from the input set to the classifier. To select the most relevant features for the classification, we ran simple linear regressions between each of the 93 LIWC features (x) and the target labels (y). We used the software R [39] to run linear regression model (lm) as below.

```
lm (y ~ x)
```

Table 3
Informative LIWC features for Author Profiling of Irony and Stereotypes: features in the Top 30 features but not in the Top 10 features. Features not in the Top 20 features are marked in asterisks(*).

Feature Name	Descriptions or Examples[4]
Word Count*	Number of words in each document
Analytical thinking*	Factor-analytically derived dimension based on several categories of function words (formal, logical, hierarchical)
Words per Sentence*	Number of words per sentence
Words > 6 letters*	Number of words with more than 6 letters
Dictionary words	LIWC dictionary word counts
Total function words	*it, to, no, very*
Total pronouns*	*I, them, itself*
Impersonal pronouns	*it, it's, those*
Articles	*a, an, the*
Common Adverbs	*very, really*
Interrogatives	*how, when, what*
Quantifiers*	*few, many, much*
Discrepancy	*should, would*
Certainty	*always, never*
Differentiation	*hasn't, but, else*
Sexual*	*horny, love, incest*
Past focus*	*ago, did, talked*
Home*	*kitchen, landlord*

where y denotes the dependent variable, which is the categorical labels coded as 0 or 1 and x denotes an independent variable, namely, one out of 93 LIWC feature vectors chosen at each time. We then calculated the R-squared value (R2) of the fitted model, which shows the proportion of the explained variance of the feature. The higher the R2, the more role the feature played in fitting the regression line. After obtaining the R2 values for each of the 93 feature vectors, we selected the features with the top 10 - 90 percent of the R2 values (in the increment of 10). We ran the classification models (SVC and RF) again using the subset of the input features.

The experiment results show that while the classification performances do not improve by subtracting certain features, the performance starts dropping when only 20 percent of the features are used (Table 2). The performances of all the other combinations were comparable; using only the top 30 percent of the features (N=27) still yields similar results to using all the features (N=93). Table 3 shows the features that were absent in the top 10 and top 20 features but were included in the top 30 features. We assume that these features are more informative than others for the irony profiling task as the classification performance started dropping significantly when these features were excluded.

The category *certainty* stands out, which is an observation aligned with general intuition: it is reasonable to assume that authors spreading irony and stereotypes would heavily rely on sentence structures using certainty (in an ironic way). Given that hyperbole is one of the tools

[4]from https://mcrc.journalism.wisc.edu/files/2018/04/Manual_LIWC.pdf

people use for being ironic [40, 41], it is expected that people use markers of certainty in the opposite direction of what they intend to say. Such linguistic patterns were commonly observed in instances correctly classified as *ironic* when we looked into the classification results as well.

- ... if you kill the opposing party you'll win **right**
- ... only use twitter to harass people since **all you dudes do is** stare at our tits yeah we got the memo...
- ... but it's **totally cool** when they call me a murderer i mean calling someone a murderer isn't abusive or anything
- ... wow a democrat telling the truth for once that would be **historic good** for you
- ... i have way more than two guns in my house and my family isn't scared or worried about them **at all**
- ... he brought his young son with him right **such a good father** ... what kind of a parent allows their kid to be used ... like this

As can be observed from the highlighted phrases, words that indicate certainty – *all*, *totally*, *at all*, *such a* – are often found in examples that were correctly classified as *ironic*. This observation aligns with the previous findings in the literature that identified hyperbolic words to be one of the markers for ironic comments [40, 41].

6. Conclusions

In this paper, we described our models for the PAN '22 Author Profiling Shared Task and reported the results. We experimented with traditional n-gram features, contextualized sentence embeddings, and lexicon-based features with traditional classifiers. We also fine-tuned transformer-based models on our dataset. The best accuracy score on our validation data was achieved from traditional classifiers that were fed lexicon-based features. We identified some of the informative features for the task by performing post-hoc feature analyses and shared some insights about the usage of irony and stereotype spreading through qualitative analyses.

References

[1] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl, R. Ortega-Bueno, P. Pezik, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, E. Zangerle, Overview of PAN 2022: Authorship Verification, Profiling Irony and Stereotype Spreaders, and Style Change Detection, in: M. D. E. F. S. C. M. G. P. A. H. M. P. G. F. N. F. Alberto Barron-Cedeno, Giovanni Da San Martino (Ed.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), volume 13390 of *Lecture Notes in Computer Science*, Springer, 2022.

[2] O.-B. Reynier, C. Berta, R. Francisco, R. Paolo, F. Elisabetta, Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO) at PAN 2022, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022.

[3] A. Reyes, P. Rosso, D. Buscaldi, From Humor Recognition to Irony Detection: The Figurative Language of Social Media, Data & Knowledge Engineering 74 (2012) 1–12. URL: https://linkinghub.elsevier.com/retrieve/pii/S0169023X12000237. doi:10.1016/j.datak.2012.02.005.

[4] D. Wilson, The pragmatics of verbal irony: Echo or pretence?, Lingua 116 (2006) 1722–1743. URL: https://linkinghub.elsevier.com/retrieve/pii/S0024384106001124. doi:10.1016/j.lingua.2006.05.001.

[5] A. Partington, Phrasal irony: Its form, function and exploitation, Journal of Pragmatics 43 (2011) 1786–1800. URL: https://linkinghub.elsevier.com/retrieve/pii/S037821661000367X. doi:10.1016/j.pragma.2010.11.001.

[6] A. Reyes, P. Rosso, Making objective decisions from subjective data: Detecting irony in customer reviews, Decision Support Systems 53 (2012) 754–760. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167923612001388. doi:10.1016/j.dss.2012.05.027.

[7] S. Attardo, Irony as relevant inappropriateness, Journal of Pragmatics 32 (2000) 793–826. URL: https://linkinghub.elsevier.com/retrieve/pii/S0378216699000703. doi:10.1016/S0378-2166(99)00070-3.

[8] A. Joshi, V. Tripathi, P. Bhattacharyya, M. J. Carman, Harnessing Sequence Labeling for Sarcasm Detection in Dialogue from TV Series 'Friends', in: Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 146–155. URL: http://aclweb.org/anthology/K16-1015. doi:10.18653/v1/K16-1015.

[9] E. Riloff, A. Qadir, P. Surve, L. D. Silva, N. Gilbert, R. Huang, 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, 2013, pp. 704–714.

[10] S. Lukin, M. Walker, Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue, in: Proceedings of the Workshop on Language in Social Media, Association for Computational Linguistics, 2013, pp. 30–40.

[11] D. I. H. Farías, V. Patti, P. Rosso, Irony detection in twitter: The role of affective content, ACM Transactions on Internet Technology (TOIT) 16 (2016) 1–24.

[12] R. González-Ibánez, S. Muresan, N. Wacholder, Identifying sarcasm in twitter: a closer look, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 581–586.

[13] A. Khatri, P. P, Sarcasm detection in tweets with BERT and GloVe embeddings, in: Proceedings of the Second Workshop on Figurative Language Processing, Association for Computational Linguistics, Online, 2020, pp. 56–60. URL: https://aclanthology.org/2020.figlang-1.7. doi:10.18653/v1/2020.figlang-1.7.

[14] A. Ghosh, T. Veale, Fracking sarcasm using neural network, in: Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics, San Diego, California, 2016, pp. 161–169. URL: https://aclanthology.org/W16-0425. doi:10.18653/v1/W16-0425.

[15] R. Akula, I. Garibay, Explainable Detection of Sarcasm in Social Media, in: Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and

Social Media Analysis, Association for Computational Linguistics, 2021, pp. 34–39. URL: https://aclanthology.org/2021.wassa-1.4.

[16] A. Avvaru, S. Vobilisetty, R. Mamidi, Detecting Sarcasm in Conversation Context Using Transformer-Based Models, in: Proceedings of the Second Workshop on Figurative Language Processing, Association for Computational Linguistics, Online, 2020, pp. 98–103. URL: https://www.aclweb.org/anthology/2020.figlang-1.15. doi:10.18653/v1/2020.figlang-1.15.

[17] S. Ilić, E. Marrese-Taylor, J. Balazs, Y. Matsuo, Deep Contextualized Word Representations for Detecting Sarcasm and Irony, in: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2–7. URL: http://aclweb.org/anthology/W18-6202. doi:10.18653/v1/W18-6202.

[18] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 11, 2017, pp. 512–515.

[19] Z. Waseem, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on twitter, in: Proceedings of the NAACL student research workshop, 2016, pp. 88–93.

[20] M. Samory, I. Sen, J. Kohne, F. Flöck, C. Wagner, Call me sexist, but...: Revisiting sexism detection using psychological scales and adversarial samples, in: Intl AAAI Conf. Web and Social Media, 2021, pp. 573–584.

[21] T. Ceron, C. Casula, Exploiting Contextualized Word Representations to Profile Haters on Twitter—Notebook for PAN at CLEF 2021, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021. URL: http://ceur-ws.org/Vol-2936/paper-160.pdf.

[22] P. Parikh, H. Abburi, P. Badjatiya, R. Krishnan, N. Chhaya, M. Gupta, V. Varma, Multi-label categorization of accounts of sexism using a neural framework, 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, 2019, pp. 1642–1652. URL: https://aclanthology.org/D19-1174. doi:10.18653/v1/D19-1174.

[23] M. Anusha, H. Shashirekha, An ensemble model for hate speech and offensive content identification in indo-european languages., in: FIRE (Working Notes), 2020, pp. 253–259.

[24] S. Zimmerman, U. Kruschwitz, C. Fox, Improving hate speech detection with deep learning ensembles, in: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018), 2018.

[25] R. O. BUENO, B. CHULVI, F. RANGEL, P. ROSSO, E. FERSINI, PAN 22 Author Profiling: Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO), 2022. URL: https://doi.org/10.5281/zenodo.6514916. doi:10.5281/zenodo.6514916.

[26] A. Avvaru, S. Vobilisetty, R. Mamidi, Detecting Sarcasm in Conversation Context Using Transformer-Based Models, in: Proceedings of the Second Workshop on Figurative Language Processing, Association for Computational Linguistics, 2020, pp. 98–103. URL: https://www.aclweb.org/anthology/2020.figlang-1.15. doi:10.18653/v1/2020.figlang-1.15, event-place: Online.

[27] S. Ilić, E. Marrese-Taylor, J. Balazs, Y. Matsuo, Deep Contextualized Word Representations for Detecting Sarcasm and Irony, in: Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Association for Computational Linguistics, 2018, pp. 2–7. URL: http://aclweb.org/anthology/W18-6202. doi:10.18653/v1/W18-6202, event-place: Brussels, Belgium.

[28] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: http://arxiv.org/abs/1908.10084.

[29] J. Pennington, R. Socher, C. Manning, Glove: Global Vectors for Word Representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1532–1543. URL: http://aclweb.org/anthology/D14-1162. doi:10.3115/v1/D14-1162.

[30] J. W. Pennebaker, R. L. Boyd, K. Jordan, K. Blackburn, The development and psychometric properties of LIWC2015, Technical Report, 2015.

[31] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of NAACL-HLT 2019, Association for Computational Linguistics, 2019, pp. 4171–4186. URL: http://arxiv.org/abs/1810.04805.

[32] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, ArXiv abs/1910.01108 (2019).

[33] J. Mao, W. Liu, A BERT-based Approach for Automatic Humor Detection and Scoring, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019), 2019, pp. 197–202.

[34] N. Babanejad, H. Davoudi, A. An, M. Papagelis, Affective and Contextual Embedding for Sarcasm Detection, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 225–243. URL: https://www.aclweb.org/anthology/2020.coling-main.20. doi:10.18653/v1/2020.coling-main.20.

[35] T. Shangipour ataei, S. Javdan, B. Minaei-Bidgoli, Applying Transformers and Aspect-based Sentiment Analysis Approaches on Sarcasm Detection, in: Proceedings of the Second Workshop on Figurative Language Processing, Association for Computational Linguistics, Online, 2020, pp. 67–71. URL: https://www.aclweb.org/anthology/2020.figlang-1.9. doi:10.18653/v1/2020.figlang-1.9.

[36] Z. Zhang, Y. Wu, H. Zhao, Z. Li, S. Zhang, X. Zhou, X. Zhou, Semantics-aware bert for language understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 9628–9635.

[37] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[38] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019.

doi:10.1007/978-3-030-22948-1_5.
[39] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2021. URL: https://www.R-project.org/.
[40] C. Burgers, M. Van Mulken, P. J. Schellens, Verbal irony: Differences in usage across written genres, Journal of Language and Social Psychology 31 (2012) 290–310.
[41] D. Ghosh, A. R. Fabbri, S. Muresan, Sarcasm analysis using conversation context, Computational Linguistics 44 (2018) 755–792.