Irony and Stereotype Spreading Author Profiling on Twitter using Machine Learning: A BERT-TFIDF based Approach

Amit Das, Nilanjana Raychawdhary, Gerry Dozier and Cheryl D. Seals

Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, USA

Abstract

In this paper we introduce our system for the task of determining whether an author spreads Irony and Stereotype in English tweets or not, a part of PAN 2022 (IROSTEREO) task. For the irony spreading author classification task, 600 authors each containing 200 tweets have been used. The uniqueness of the task is that it is not a classification between ironic and non ironic tweets, instead it is a classification of irony and non irony spreading authors. The task contains a subtask also that addresses stereotype stance detection. For the previous years, several representation methods like character/word n-grams etc. have been used for tweet representations, but there was not a clear clue whether a combination of other representations would be helpful. To do this end, we introduce BERT combined with TFIDF representation to address this specific problem. Later we used Logistic Regression classifier for the classification task. It was seen that the BERT representation combined with TFIDF showed very promising results.

Keywords

Irony detection, Author profiling, Natural language processing, Twitter data

1. Introduction

Irony is a deeply pragmatic and diverse linguistic phenomena that has been thoroughly explored in numerous domains [1]. Irony detection has recently gained a lot of interest in the machine learning and NLP world due to the high frequency of sarcastic expressions in social media [2]. In the context of sentiment analysis, their language collocation has a tendency to flip polarity, making machine-based irony identification important [3] [4]. The goal of irony detection is to develop computational algorithms that automatically recognize this phenomena in written languages [5] [6] [7]. Several researchers have attempted the irony detection problem, according to the literature [8] [9]. Many of these efforts have been devoted to the examination of the textual representation and features [5] [6] [10]. PAN 2022's Irony Detection job focuses on characterizing irony and stereotype spreaders on Twitter. The work focuses on characterizing ironic authors on Twitter, with a focus on authors that use irony to disseminate prejudices about women or the LGBT community, for example [11] [12]. The task's purpose is to categorize authors as ironic or not based on how many tweets they have with ironic content [13]. A subgroup of those authors is studied who use irony to express stereotypes in order to see if

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

azd0123@auburn.edu (A. Das); nzr0044@auburn.edu (N. Raychawdhary); doziegv@auburn.edu (G. Dozier); sealscd@auburn.edu (C. D. Seals)

state-of-the-art models can differentiate these cases as well. As a result, given a list of Twitter users and their tweets, the purpose is to identify those who can be classified as ironic. A subtask of the task deals with Stereotype Stance Detection. Ironic authors have used stereotypes to harm the target (such as immigrants) or to support it in some way. This subtask's objective is to determine whether sarcastic authors are using stereotypes to support or undermine the target. The objective is to identify their general viewpoint given the subset of sarcastic authors who used stereotypes in some of their tweets. The TIRA platform has been used by the participants to assess their approaches. This platform can be used to deploy and test applications automatically [14]. The algorithms are assessed using a common test dataset, the same metrics, as well as the amount of time required to generate the response.

Irony is a profoundly pragmatic and versatile linguistic phenomenon. As its foundations usually lay beyond explicit linguistic patterns in reconstructing contextual dependencies and latent meaning, such as shared knowledge or common knowledge [2], automatically detecting it remains a challenging task in natural language processing. In this paper, we use two representations: 1) BERT and 2) Term Frequency Inverse Document Frequency (TFIDF) combined with BERT to address this issue. Later the classification task was implemented using Logistic Regression classifier.

2. Related Work

Irony detection is a very challenging task that encountered a lot of development through the years. Here are some of the recent research works that contribute to the problem.

Identifying the important components to recognize irony in English customer evaluations has been the attention of Reyes and Rosso [15]. To reflect irony, they used six categories in their model: n-grams, POS ngrams, funny profiling, positive/negative profiling, affective profiling, and pleasantness profiling. Customers' online reviews were chosen as part of the dataset [16]. They employed three distinct classifiers to reach their results, which showed very competitive performance.

The automatic detection of irony was framed as a classification problem by Barbieri and Saggion [17]. They created a model that could detect irony in the social network Twitter using linguistic variables like frequency, written/spoken contrasts, attitudes, ambiguity, intensity, synonymy, and structure. Nayel et al. [16] picked tweets with the hashtag irony and a few other subjects to generate a linguistically motivated set of features. Their model outperformed the bag-of-words technique across domains, according to the findings.

Teh et al.'s [18] investigation focused on the use of coarse language for the detection of hate speech. Based on the use of profanity, the writers divided 500 YouTube comments into 8 different categories of hate speech. Numerous other studies of a similar nature focused on identifying hate speech [18] [19], social media abuse [20] [21], fake news on Twitter [22], and cyberbullying [19]. On author profile based on their tweets, numerous publications and shared tasks are available [22] [23] [24].

A model for irony detection in Twitter emotIDM [25] was developed by formulation of the task as a classification problem. It was evaluated on a set of representative Twitter corpora that included samples of ironic and non ironic messages, which were different along various dimensions like size, balanced vs imbalance distribution, collection methodology and criteria [16]. Results showed good performances in classification.

KLUEnicorn [26] offered a system that used a Naive Bayes classifier to build word embeddings using several adverb categories and named entities, as well as semantic and lexical data. Various supervised classification techniques, such as Randomizable Filtered Classifier (RFC), Bayesian Network (BayesNet), IBk, and others, were reviewed and compared in another comprehensive review [27].

In order to increase Stereotype Stance Detection, Mohammad et al. [28] looked into the significance of utilizing the sentiment that is expressed in a text. Without taking into account the target, the total sentiment expressed in each occurrence was annotated in the SemEval-2016 Task 6 dataset. They used n-grams, char-grams, sentiment features from many lexica, including the Hu and Liu lexicon [29], EmoLex [30], and the MPQA Subjectivity Lexicon [31]. Additionally, they took into account the target of interest in the tweet's appearance or absence, as well as the frequency of part-of-speech tags, emoticons, hashtags, uppercase letters, lengthened phrases, and punctuation. They were able to outperform the competition by combining these features with a support vector machine classifier.

In order to forecast the authors' ages using the Maximum Entropy classifier and LASSO regression, Hong et al. [32] combined numerous datasets, including Fisher English transcript and Blog authorship, to create a dataset with a variety of stylistic and content-based variables. With the exception of increased age limits, both models produced good outcomes.

To detect users with different perspectives in regards to stereotype stance detection in tweets, Rajadesingan and Liu [33] employed a semi-supervised framework in conjunction with a supervised classifier. The authors took advantage of a retweet-based label propagation theory, which is based on the fact that if a lot of users retweet a specific pair of tweets in a reasonable amount of time, it is quite likely that the two tweets are related in some way. Based on how closely a tweet aligns with the ideals of the labels surrounding it, they categorized it as "for" or "against" in their study.

A label propagation technique was employed for community discovery in the work of Raghavan et al. [34]. Their method was exceptionally straightforward and effective; in fact, each node adopted the label that the majority of its immediate neighbors now have in their iterative procedure, and it appeared to perform exceptionally well in unsupervised environments.

In order to get meaningful phrase embeddings, there are new methods for fine-tuning language models [35] [36]. Using the universal sentence encoder, TFIDF, and a support vector machine for the case law retrieval challenge in the last COLIEE edition, Rabelo et al. [37] outperformed many of the models. Therefore, we expect that TFIDF in conjunction with BERT representation could also be effective for the task of identifying irony authors. The next section explains the datasets used in this research.

3. Dataset

For the irony detection task, the dataset contained tweets of 600 authors each having 200 tweets. It was split into two categories: 1) validation dataset containing tweets of 420 authors and 2) test dataset containing tweets of 180 authors. The validation dataset is a balanced dataset (50%

of them were irony and 50% of them were non irony) containing total 84000 tweets (420 authors each having 200 tweets). This dataset is used for the training purposes. For testing, 180 authors each containing 200 tweets are used. The training set is balanced, i.e. out of 420 authors 210 are irony and 210 are not irony. The details of the dataset is shown in Table 1.

Table 1
Irony and Non irony spreading author Dataset

Data split	No. of authors	No. of tweets of each author
Training	420	200
Testing	180	200

For the Stereotype Stance Detection subtask, the dataset contained tweets of 200 authors each having 200 tweets. It was again split into two categories: 1) validation dataset containing tweets of 140 authors and 2) test dataset containing tweets of 60 authors. The validation dataset is an imbalanced dataset containing 28000 tweets (140 authors each having 200 tweets). This dataset is used for the training purposes. For testing, 60 authors each containing 200 tweets are used. The goal of this subtask is to detect the stance of how stereotypes are used by ironic authors, if in favour or against the target. Table 2 shows the details of the dataset. The training set is imbalanced, i.e. out of 140 authors 94 are AGAINST and 46 are INFAVOR.

Table 2
Stereotype spreading author Dataset

Data split	No. of authors	No. of tweets of each author
Training	140	200
Testing	60	200

4. Methods

In this work we implement the following method for tweet representation: BERT combined with TFIDF. We detail each of the feature spaces in the following lines:

4.1. BERT

In this section, we'll go over BERT and how to use it in depth. The design of a neural encoder for natural language sequences has been changed by Transformer [38], a sequence transduction model based on attention mechanisms. The transformer architecture allows sequential data to be learned. To improve on largely unidirectional language model training, Devlin et al. [39] developed Bidirectional Encoder Representations from Transformers (BERT). BERT makes deep bidirectional language encoding training achievable by employing the masked language modeling (MLM) loss [40]. BERT employs next-sentence prediction (NSP), an extra loss for pre-training that aims to learn high-level linguistic coherence by predicting whether or not two text segments should come sequentially in the original text [40].

The sentence had to be tokenized first before the embeddings could be created. Point to be noted, BERT can only handle sentences with a length of 512 tokens or less. BERT's authors advise using the BERT Base Uncased model in the majority of cases unless it is clear that using a case-sensitive model will be beneficial to the task [41]. Using 1s and 0s to discriminate between the two sentences, BERT is trained on and anticipates sentence pairs [41]. That is to say, we must indicate whether each token in "tokenized text" fits in sentence 0 (a string of 0s) or sentence 1 (a series of 1s). We constructed a vector of 1s for each token in our input sentence since single-sentence inputs just need a string of 1s for our needs [41].

We then called the BERT model after converting our data to torch tensors. The number of layers (13 layers), the batch number (1 sentence), the word/token number (22 tokens in our sentence), and the hidden unit/feature number make up the complete set of hidden states for this model (768 features). The first element represents the input embeddings, and the remaining elements are the outputs of each of the 12 layers of BERT, hence the layer number is 13 [41].

When sending several sentences to the model at once, the batch size, the second dimension, is employed. There would be one batch total. We had 13 distinct vectors, each of which was 768 bytes long, for each token in our input. We combined the final four layers to produce a word vector with a length of 3072 (4×768 = 3072) per token. We calculated the average of the second-to-last hidden layer of each token to produce a single vector of 768 length for our complete text[41].

4.2. TFIDF

We apply the well-known Term Frequency Inverse Document Frequency (TFIDF) weighting system in our methodology to extract traditional features. TFIDF is a combination of two different terms: Term Frequency (TF) and Inverse Document Frequency (IDF) [42]. The term TF is used to calculate the frequency of a term in a document [43]. The term frequency for a term t and a document d is defined by

$$tf_{d,t} = \frac{n_{d,t}}{|d|} \tag{1}$$

where $n_{d,t}$ is the number of occurrences of the term t in the document d. The term frequency $tf_{d,t}$ is then the number of occurrence of the term t in document d divided by the total number of tokens in the document.

The inverse document frequency of a term t in the whole collection is,

$$idf_t = log\frac{|D|}{|d : t \in d|} \tag{2}$$

where |D| is the number of classes in the classification problem and $|d : t \in d|$ is the number of document(s) where the term t appears.

When calculating a document's term frequency, it can be seen that the algorithm evaluates all keywords similarly, regardless of whether they are stop words or not, which is incorrect because all keywords have varying relevance [43]. The inverse document frequency method gives less weight to often occurring words and more weight to infrequently occurring terms [43]. Mathematically, TFIDF is the multiplication of term frequency (TF) and inverse document frequency (IDF). The formula that is used to compute the TFIDF of term t present in document d is:

$$tfidf_{d,t} = tf_{d,t} * idf_t = \frac{n_{d,t}}{|d|} * log\frac{|D|}{|d : t \in d|} \tag{3}$$

TFIDF's purpose is to lessen the impact of less informative tokens that appear frequently in a data corpus [44]. We used TfidfVectorizer features from scikit-learn to perform the TFIDF task [45]. Table 3 shows the TFIDF parameter values used for the tasks.

Table 3
TFIDF parameters

Task	TFIDF_max_df	TFIDF_min_df
Irony spreading author profiling	0.70	1
Stereotype spreading author profiling	0.95	1

4.3. BERT combined with TFIDF

Sentence-BERT, which surpasses the current embedding techniques and is deemed effective for numerous downstream applications, was introduced by Reimers et al. [36]. TFIDF is used to evaluate how relevant a word is to a document in a collection of documents. The TFIDF score can be fed to the Bert model to improve the predicting performance. In order to produce a deeper and more insightful quantitative representation of the data, we used this embedding approach. We combined TFIDF with word embedding. We put a threshold of <1000 words while implementing the TFIDF. The idea is to preserve the grammatical regularities in each document intact.

4.4. Classifier

The Logistic Regression classifier is used in order to classify the irony and stereotype spreading authors. Logistic Regression uses logistic function to model binary dependent variable. The equation can be given as:

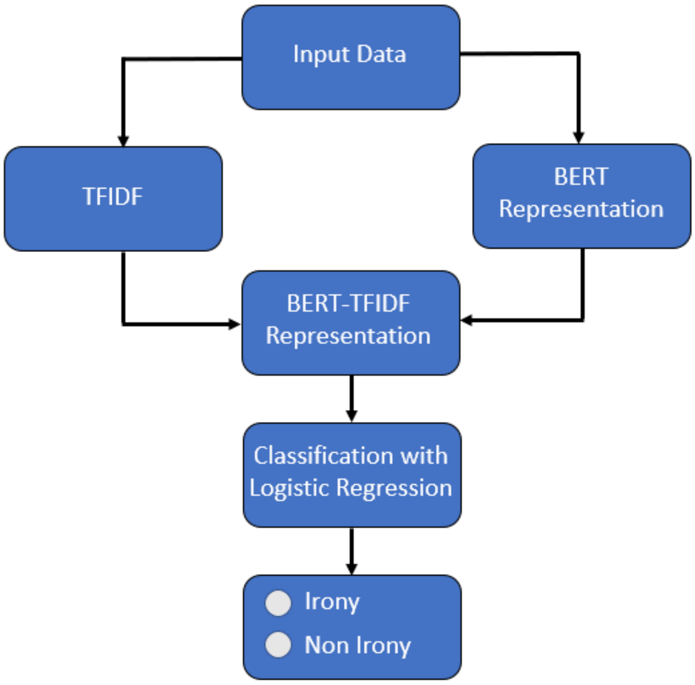
$$P = \frac{e^{a+bx}}{1 + e^{a+bx}} \tag{4}$$

We used LogisticRegression class from scikit-learn library to implement Logistic Regression model [46].

Figure 1 shows the architecture of our proposed model. After splitting the dataset into training and testing set, they are first tokenized using BERT representation. Then the TFIDF is combined with it to make the training dataset richer. Lastly the training and the predictions are made using Logistic Regression classifier.

For the vectorization, every word is assigned a unique number. Each data is transformed into an N-dimensional vector, where N is a number of words in the data.

Figure 1: Architecture of the Proposed Method


Input Data
TFIDF
BERT Representation
BERT-TFIDF Representation
Classification with Logistic Regression
Irony
Non Irony

In the following section we will explain the evaluation results of all the models used on validation dataset and the test dataset.

5. Results & Discussion

To understand the efficiency of our models, first we split the validation data into training and testing set. Out of 420 authors, we used 336 of them as training and remaining 84 were used for testing. The datasets were split balanced, i.e. for both the training and testing data, 50% of them were irony and 50% of them were non irony. We combined all 200 tweets of each person and treated it as a single string. It basically created 420 strings for 420 authors each string containing the combination of 200 tweets of each author. Initially we were only concerned about the accuracy of different machine learning models for only the BERT representation. The strings were then converted to numeric values using the BERT. We implemented the following five machine learning algorithms to check the best accuracy: KNN, SVM, Decision Tree, Naive Bayes and Logistic Regression and it was seen that the Logistic Regression was proved to be the best in terms of efficiency and accuracy. To measure the accuracy of an algorithm, we used the formula in equation 5.

$$Accuracy = \frac{TrueLabel}{TrueLabel + FalseLabel} \tag{5}$$

Where True Label refers to correct prediction and False Label refers to incorrect classification. The classification results obtained from the five algorithms on the validation dataset are given below:

Table 4
ML algorithms implemented on ironic author profiling validation dataset

Algorithm	Accuracy(%)
KNN	89.2
SVM	88
Decision Tree	77.3
Naive Bayes	91.6
Logistic Regression	92.8

The BERT implementation method was then implemented on the testing dataset of tweets of 180 authors after training it with the validation dataset tweets of 420 authors. It was seen that the classification results were not very promising. However, after combining the BERT representation with TFIDF, the classification results were improved from 38% to 67% which was an increase of around 76%. The classification results on the test dataset are shown in Table 5.

Table 5
Accuracy on ironic author profiling test dataset

Representation	Classifier	Accuracy(%)
BERT	Logistic Regression	38
BERT-TFIDF	**Logistic Regression**	**67**

The similar method was implemented to address the subtask of stereotype stance detection used by ironic authors either in favour or in against. Unlike the dataset used for ironic author classification, the dataset used for stereotype stance detection was smaller in size and also imbalanced. The classification result of stereotype stance detection problem using our model is shown in Table 6. We obtained an overall macro F1 score of 0.45 and F1 score of 0.19 InFavour.

Table 6
Accuracy on stereotype favouring author profiling test dataset

Representation	Classifier	Accuracy(%)
BERT-TFIDF	**Logistic Regression**	**58**

The BERT representation alone itself was not sufficient to make the classification results high. The TFIDF's purpose is to increase the impact of more informative tokens that appear frequently in a data corpus. When the TFIDF was combined with BERT representation, the accuracy of the model was significantly improved. When we compare the classification accuracies of the two tasks, the accuracy of the irony detection problem was higher than the stereotype stance detection problem probably because of larger number of training and testing dataset. It was very interesting to see the usefulness of this model on both balanced and imbalanced type of dataset.

6. Conclusion

In this paper we presented our method to PAN 2022 Profiling Irony and Stereotype Spreaders on Twitter (IROSTEREO) task to address the irony spreading author detection problem on twitter data. The task contained a subtask also that addresses Stereotype Stance Detection i.e. detecting the stance of how stereotypes are used by ironic authors, if in favour or against the target. To address both the tasks, first we implemented BERT method and then BERT combined with TFIDF for representation of the tweets. Then Logistic Regression classifier was used to classify the irony and non-irony spreading authors. The BERT method was used to convert the text data into equivalent numeric data, and the TFIDF represented the more important tokens. It was seen that combining BERT representation with TFIDF significantly improved the result. To conclude, we have shown some useful techniques for irony and stereotype spreaders classification. How this model behaves to a different type of dataset, will be a future direction to explore.

References

[1] E. Marrese-Taylor, S. Ilic, J. A. Balazs, Y. Matsuo, H. Prendinger, Iiidyt at semeval-2018 task 3: Irony detection in english tweets, arXiv preprint arXiv:1804.08094 (2018).

[2] A. Joshi, P. Bhattacharyya, M. J. Carman, Automatic sarcasm detection: A survey, ACM Computing Surveys (CSUR) 50 (2017) 1–22.

[3] S. Poria, E. Cambria, D. Hazarika, P. Vij, A deeper look into sarcastic tweets using deep convolutional neural networks, arXiv preprint arXiv:1610.08815 (2016).

[4] C. Van Hee, E. Lefever, V. Hoste, Guidelines for annotating irony in social media text, version 2.0, LT3 Technical Report Series (2016).

[5] D. Davidov, O. Tsur, A. Rappoport, Semi-supervised recognition of sarcasm in twitter and amazon, in: Proceedings of the fourteenth conference on computational natural language learning, 2010, pp. 107–116.

[6] T. Veale, Y. Hao, Detecting ironic intent in creative comparisons, in: ECAI 2010, IOS Press, 2010, pp. 765–770.

[7] 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.

[8] A. Reyes, P. Rosso, On the difficulty of automatically detecting irony: beyond a simple case of negation, Knowledge and Information Systems 40 (2014) 595–614.

[9] A. Reyes, P. Rosso, T. Veale, A multidimensional approach for detecting irony in twitter, Language resources and evaluation 47 (2013) 239–268.

[10] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, A. Reyes, Semeval-2015 task 11: Sentiment analysis of figurative language in twitter, in: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), 2015, pp. 470–478.

[11] J. Bevendorff, B. Chulvi, E. Fersini, A. Heini, M. Kestemont, K. Kredens, M. Mayerl,

R. Ortega-Bueno, P. Pęzik, M. Potthast, et al., Overview of pan 2022: Authorship verification, profiling irony and stereotype spreaders, style change detection, and trigger detection, in: European Conference on Information Retrieval, Springer, 2022, pp. 331–338.

[12] 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.

[13] 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.

[14] 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.

[15] A. Reyes, P. Rosso, Mining subjective knowledge from customer reviews: A specific case of irony detection, in: Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis (WASSA 2.011), 2011, pp. 118–124.

[16] H. A. Nayel, W. Medhat, M. Rashad, Benha@ idat: Improving irony detection in arabic tweets using ensemble approach., in: FIRE (Working Notes), 2019, pp. 401–408.

[17] F. Barbieri, H. Saggion, Modelling irony in twitter, in: Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, pp. 56–64.

[18] P. L. Teh, C.-B. Cheng, W. M. Chee, Identifying and categorising profane words in hate speech, in: Proceedings of the 2nd International Conference on Compute and Data Analysis, 2018, pp. 65–69.

[19] Y. Chen, Detecting offensive language in social medias for protection of adolescent online safety (2011).

[20] E. Chandrasekharan, U. Pavalanathan, A. Srinivasan, A. Glynn, J. Eisenstein, E. Gilbert, You can't stay here: The efficacy of reddit's 2015 ban examined through hate speech, Proceedings of the ACM on Human-Computer Interaction 1 (2017) 1–22.

[21] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, Y. Chang, Abusive language detection in online user content, in: Proceedings of the 25th international conference on world wide web, 2016, pp. 145–153.

[22] F. Rangel, A. Giachanou, B. H. H. Ghanem, P. Rosso, Overview of the 8th author profiling task at pan 2020: Profiling fake news spreaders on twitter, in: CEUR Workshop Proceedings, volume 2696, Sun SITE Central Europe, 2020, pp. 1–18.

[23] F. Rangel, P. Rosso, M. Montes-y Gómez, M. Potthast, B. Stein, Overview of the 6th author profiling task at pan 2018: multimodal gender identification in twitter, Working Notes Papers of the CLEF (2018) 1–38.

[24] B. G. Patra, K. G. Das, D. Das, Multimodal author profiling for twitter, Notebook for PAN

at CLEF (2018).

[25] D. I. Hernandez Farias, V. Patti, P. Rosso, Irony detection in twitter: The role of affective content, ACM Transactions on Internet Technology 16 (2016).

[26] L. Dürlich, Kluenicorn at semeval-2018 task 3: A naive approach to irony detection, in: Proceedings of The 12th International Workshop on Semantic Evaluation, 2018, pp. 607–612.

[27] U. B. Baloglu, B. Alatas, H. Bingol, Assessment of supervised learning algorithms for irony detection in online social media, in: 2019 1st International Informatics and Software Engineering Conference (UBMYK), IEEE, 2019, pp. 1–5.

[28] S. M. Mohammad, P. Sobhani, S. Kiritchenko, Stance and sentiment in tweets, ACM Transactions on Internet Technology (TOIT) 17 (2017) 1–23.

[29] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, pp. 168–177.

[30] S. M. Mohammad, P. D. Turney, Crowdsourcing a word–emotion association lexicon, Computational intelligence 29 (2013) 436–465.

[31] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of human language technology conference and conference on empirical methods in natural language processing, 2005, pp. 347–354.

[32] J. Hong, C. A. Mattmann, P. Ramirez, Ensemble maximum entropy classification and linear regression for author age prediction, in: 2017 IEEE International Conference on Information Reuse and Integration (IRI), IEEE, 2017, pp. 509–516.

[33] A. Rajadesingan, H. Liu, Identifying users with opposing opinions in twitter debates, in: International conference on social computing, behavioral-cultural modeling, and prediction, Springer, 2014, pp. 153–160.

[34] U. N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Physical review E 76 (2007) 036106.

[35] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al., Universal sentence encoder for english, in: Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations, 2018, pp. 169–174.

[36] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019).

[37] J. Rabelo, M.-Y. Kim, R. Goebel, M. Yoshioka, Y. Kano, K. Satoh, Coliee 2020: methods for legal document retrieval and entailment, in: JSAI International Symposium on Artificial Intelligence, Springer, 2020, pp. 196–210.

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[39] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[40] H. Choi, J. Kim, S. Joe, Y. Gwon, Evaluation of bert and albert sentence embedding performance on downstream nlp tasks, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 5482–5487.

[41] C. McCormick, N. Ryan, Bert word embeddings tutorial, URL: https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial (2019).

[42] S. Qaiser, R. Ali, Text mining: use of tf-idf to examine the relevance of words to documents, International Journal of Computer Applications 181 (2018) 25–29.

[43] A. A. Hakim, A. Erwin, K. I. Eng, M. Galinium, W. Muliady, Automated document classification for news article in bahasa indonesia based on term frequency inverse document frequency (tf-idf) approach, in: 2014 6th international conference on information technology and electrical engineering (ICITEE), IEEE, 2014, pp. 1–4.

[44] A. Gaydhani, V. Doma, S. Kendre, L. Bhagwat, Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach, arXiv preprint arXiv:1809.08651 (2018).

[45] F. Pedregosa, et al., sklearn. feature_extraction. text. tfidfvectorizer (2013).

[46] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.