Profiling Irony and Stereotype Spreaders with Encoding Dependency Information using Graph Convolutional Network

Notebook for PAN at CLEF 2022

Hamed Babaei Giglou[1,5], Mostafa Rahgouy[2], Ali Rahmati[3,5], Taher Rahgooy[4] and Cheryl D Seals[2]

[1]*Department of Computer Science, University of Tabriz, Tabriz, Iran*
[2]*Department of Computer Science, Auburn University, Alabama, USA*
[3]*Algorithms Graphs Machine Learning Lab, Amirkabir University of Technology, Tehran, Iran*
[4]*Department of Computer Science, University of West Florida, Florida, USA*
[5]*Part AI Research Center, Tehran, Iran*

Abstract

Figurative language is using words in a way that deviates from the conventional order and meaning in order to ask the reader or listener to understand the meaning by virtue of its relation to some other meaning or concept. It is a rapidly growing area in Natural Language Processing, including the processing of irony, sarcasm, as well as other figures. With irony, language is employed in a figurative and subtle way to mean the opposite of what is literally stated. In this paper, we hypothesize that encoding dependency information allows us to find word associations that capture the author's style where the main intent is to spread stereotypes in the form of irony. To do this end, we proposed a graph convolutional network that is able to learn the heterogeneous text graphs, representing various dependency information in the text, which enables classifiers to classify ironic and non-ironic spreaders via looking at heterogeneous features. Experimental results are very promising and show the effectiveness of our proposed method.

Keywords

Graph Neural Network, Graph Convolutional Network, Graph Attention Network, Irony Detection, Stereotype

1. Introduction

The raising of computational social science in social media challenges modern computational linguistics and text analytics. The challenge concerns the advancement of natural language processing (NLP) methodologies toward the transformation of manuscripts into structured data for the identification of special text characteristics, such as irony and stereotypes content.

CLEF 2022 – Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
h.babaei98@ms.tabrizu.ac.ir (H. Babaei Giglou); mzr0108@auburn.edu (M. Rahgouy); ali.rahmati@partdp.ai (A. Rahmati); trahgooy@students.uwf.edu (T. Rahgooy); sealscd@auburn.edu (C. D. Seals)
https://hamedbabaei.github.io (H. Babaei Giglou); http://www.rahgooy.com/ (T. Rahgooy); https://www.eng.auburn.edu/~sealscd/ (C. D. Seals)
0000-0002-2889-0522 (T. Rahgooy)

Figurative language (FL) seems ubiquitous in all social media discussion forums and chats, posing extra challenges for linguistics. The main FL expression forms are sarcasm, irony, and stereotypes. With irony, language is employed in an FL way to mean the opposite to what is stated, which in the matter of sarcasm (a more aggressive type of irony) the intent is to scorn a viction. On the other hand, stereotypes are often used in discussions about controversial issues such as immigration or sexism and misogyny. The challenging part is the user behaviors in social media since users tend to violate common grammar and vocabulary rules and even use various FL forms to communicate their message. In past years emergence of research in this area got lots of interest and this year, at the Author Profiling (AP) task [1] in PAN [2] the emphasis will be given to those authors that employ irony to spread stereotypes. The task has been defined as follows:

> Given a Twitter feed in English, determine whether its author spreads Irony and Stereotypes.

Where input data is timelines of authors sharing Irony and Stereotypes towards, for instance, women or the LGTB community[1]. The major concern is to identify authors that spread ironic content. The task was designed to consider a subset of authors that employs irony to convey stereotypes in order to investigate the state-of-the-art models capability in this domain. So, the given authors tweets, and the proposed models should profile those that can be considered ironic spreaders.

The rest of the paper is organized as follows. Section 2 presents related works. Section 3 describes the method. Section 4 describes the experimental setups such as dataset, metrics, and training setups. Next, in section 5 we discussed results and analysis. Finally, section 6 presents our conclusions.

2. Related Works

In recent years AP considers the major topics in social media analysis topics such as bot-gender [3], fake news [4], hate speech [5], and this year irony identifications. The difference between AP tasks from others is the nature of AP is to identify authors that trying to spread figurative content. Each year, authors have done major preprocessing, feature engineering, and classifications. Due to the complex nature of the task, and to avoid any bias, organizers introduced balanced datasets to keep participants away from the challenge of imbalanced classification problems and allow them to focus on the problem itself. To this end, most of the authors heavily relied on preprocessing and representation parts to overcome the challenge. The representations mostly included such as n-grams, stylistics, personality, and deep learning-based feature engineering methods such as embeddings to extract high-level features to feed machine learning or deep learning models. In previous years, word/character level n-grams representations with SVM classifier and LDSE [6] showed us a strong baseline. Analyzing these models and participant approaches reveals that word occurrences are playing a vital role in solving these challenges rather than the semantic aspect of the task. Many approaches are proposed for AP but no one tried Graph Neural Networks (GNNs) [7] to capture the word and authors associations. In this

[1]Also known as the LGBTQ+ community, GLBT community, or the gay community

work, we proposed a model to learn the graph representations to identify irony and stereotype spreaders on Twitter.

Stereotypes are a type of social bias increasingly present in human interaction in social networks. This challenging task is being studied by computational linguistics because of the rise of hate messages, offensive language, sarcasm, and discrimination that many people receive. From the sarcastic argument, in [8] work, the authors proposed a hybrid CNN-BiLSTM model to capture the stylistic information via statistical and contextualized features. Their analysis reveals that word co-occurrences play a vital role in detecting sarcastic contents. In hate messages, [9] showed the effectiveness of word/char n-gram features with LDSE in capturing co-occurrence features. In addition to co-occurrence representation, the contextualized features showed to be effective in hate speech spreaders detection. Moreover, in offensive language, at toxic span detection task [10] proposed an ensemble feature to enrich the representation for detecting phrases that make context toxic. All of the research tried to enrich the representations to achieve high-quality features for classifications. However, in profiling authors in AP tasks in social media, no one considered heterogeneous features using graphs except in similar work to our work, the [11] introduced graph convolutional networks (GCNs) for abusive content profiling. They showed that heterogeneous graph-structured modeling of communities significantly advances the current state-of-the-art in abusive language detection.

The [12] proposed two different approaches for identifying stereotypes in social media, a deep learning model based on Transformers; and a text masking technique that has been recognized for its capabilities to deliver good and human-understandable results. Moreover, in [13] a task from SemEval 2019 studied hate speech content against immigrants and women in English and Spanish. The [14] studied affective content role in irony identification at Twitter and concluded that according to classification experiments over different corpora show that affective information helps in distinguishing among ironic and non-ironic tweets.

3. Method

In this section, we describe the details of our proposed model. Our proposed approach aims to predict whether the user is keen to spread irony in form of a stereotype or not. Recently, many researchers focused on examining whether graph convolutional networks (GCNs) [15] could handle different NLP tasks, especially text classification. Inspired by [16] we believed applying GCNs to irony and stereotype detections is well-studied because of dependency information. GCN is a multi-layer neural network generalized from Convolutional Neural Networks (CNNs), which directly operates on the graph-structured data and learns representation vectors of nodes based on the properties of their neighborhoods. We hypothesize that the author's word usage plays a kind of fingerprinted feature that may convey a meaning when it is connected to words. Similar connections may connect the ironic spreaders in a graph. So, connection represents the dependency information that allows capturing valuable embeddings to represent irony and stereotype spreaders. Considering this we represent the author's tweets in form of graph embeddings to learn the linguistic behavior of the authors to form graph embeddings as dependency information through nodes (words, authors) and edges (the connection between words). Figure 1 depicts an overview of our proposed method. First, we combined each author's tweets, then we perform the preprocessing. Next, we construct the graph and fed it into the GCN architecture. In the following, we described each component in detail.

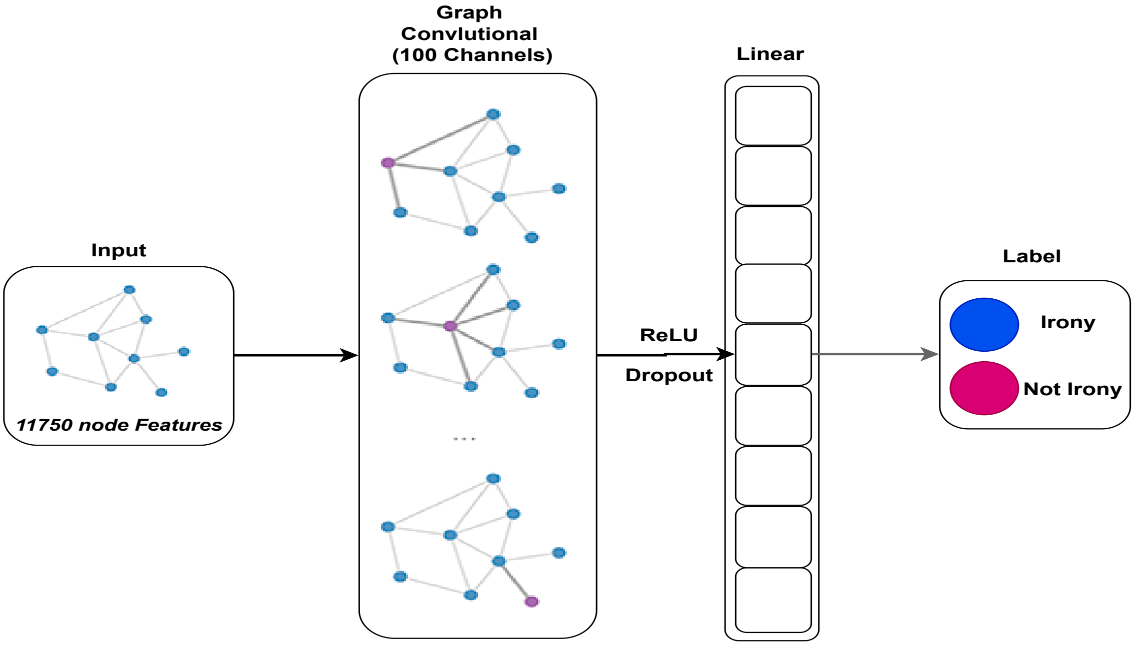

Figure 1: Architecture of Proposed Method

3.1. Preprocessing

Since we want to construct the graph and more words (nodes) lead to a higher number of edges, as a result of this we had to use higher RAM for training. To avoid this problem, we performed the following steps:

- Removing punctuations and special characters.
- Lower the text.
- Applying word frequency threshold fq_w (words with less than this frequency will be removed from vocabs).
- Removing stopwords.
- Removing URLs, hashtags, mentions, and reserved words (RT, FAV)

In the end, the graph was constructed with $fq_w = 15$ to use preferable RAM in graph construction and learning.

3.2. Graph Convolutional Network

The graph data structure which consists of vertices and edges can be represented as follows:

$$G = (V, E)$$

Where V is the vertices and E is the edges. For graph learning, the input should be in form of graph data structure. Inspired by a GCN, TextGCN [17] we construct an entire corpus-based graph, which uses all the words and authors in the corpus as graph nodes and sets the word-word and word-author edges to preserve the global word co-occurrence and word-authors relations in the graph structure. This representation ables the extract dependency features and the author's overlap and sensitiveness to most informative words. Then, it would be modeled by GCN learning for irony spreaders detection. The construction of the graph and GCN learning architecture is described in the following sections.

3.2.1. Graph Node Construction

To construct nodes V, we used all the words and authors in the dataset. The number of nodes equals:

$$|V| = N = A + M$$

Where N is the number of nodes, A is the number of users, and M is the number of unique words in the dataset. For initial node representation, we used two types of representations: 1) Sentence-BERT, and 2) one-hot embeddings.

- Aim of Sentence-BERT[18] is to obtain embeddings for author and word nodes. In this setting, the Sentence-BERT adds a pooling operation to the output of DistilRoBERTa[19] to derive a fixed-sized sentence embedding by computing the mean of all output vectors. In the end, we obtained $N = 11,750$ nodes, and a node representation matrix with the shape of $[11,750,768]$.
- In one-hot embedding construction we simply set one-hot feature matrix $X = I$ as an identity matrix which means every word or author is represented as a one-hot vector.

The one-hot embedding representation is used for edge construction and edge weight calculations.

3.2.2. Graph Edge Construction

We build a large and heterogeneous text graph that contains word nodes and document nodes so that global word co-occurrence can be explicitly modeled. We utilize author-word and author-author edges in addition to word-word edges. We build edges among nodes based on word occurrence in the author's tweets (author-word edges), word co-occurrence in the whole dataset (word-word edges), and authors word co-occurrences in the whole dataset (author-author edges). We used TFIDF vectors to weight author-words edges. Next, we employ point-wise mutual information (PMI), a popular measure for word associations, to calculate weights between word-word nodes. Also, we have considered the Jaccard similarity score based on TFIDF vectors to calculate weights between author-author edges. Formally, the weight of the edge between node i and node j is defined as follows:

$$A_{i,j} = \begin{cases} PMI(i,j) & i,j \text{ are words, } PMI(i,j) > 0 \\ TFIDF_{ij} & i \text{ is author, } j \text{ is word} \\ 1 & i = j \\ Jaccard(i,j) & i,j \text{ are authors} \\ 0 & \text{otherwise} \end{cases}$$

The PMI value of word pair i, j and Jaccard value of author pair i, j are calculated as:

$$PMI(i,j) = log\frac{P(i,j)}{P(i)P(j)} \quad , \qquad P(i,j) = \frac{\#W(i,j)}{\#W} \quad , \qquad P(i) = \frac{\#W(i)}{\#W}$$

$$Jacard(i,j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|}$$

where $\#W(i)$ is the number of sliding windows in a data that contain the word i, $\#W(i,j)$ is the number of the sliding windows that include both word i and j, and $\#W$ is the total number of sliding windows in the dataset. With $PMI(i,j) > 0$, only the high semantic correlation of words in the dataset has been considered. Also in Jaccard similarity equation, the A_i and A_j are set of words in author i and j tweets, respectively.

In PMI we have considered a sliding window of 20, where at each window the word co-occurrences will be updated for PMI calculation. Also, for author-author edges, we used a threshold of 0.2 which means Jaccard similarity of higher than this threshold will be considered as edges. In the end, we obtained $12,935,016$ edges to feed the GCN model.

3.2.3. Graph Learning

To build a corpus-level GCN-based text classification model, after building the text graph, we feed the graph into a simple one-layer GCN with 100 channels. Next, a ReLU activation function and Dropout with a probability of 0.5 are applied to outputs of one-layer GCN. After that, outputs are fed into a linear layer with an input size of 100 and output size as the labels set. Next, outputs are fed into a softmax classifier. The Adam optimizer with a learning rate of 0.001 and an epoch number of 1000 was chosen for the training of the model.

4. Experimental Setups

4.1. Dataset

Table 1 presents the statistics of the dataset[2] that consists of 600 authors for English with ironic and none ironic labels. For each author at least 200 Tweets were collected. The dataset is balanced, with 300 authors for each class. Dataset has split into training and test sets, following the $70/30$ proportion. For modeling, we combined each author's tweets as a single text to feed models.

[2]https://zenodo.org/record/6514916

Language	Traning	Testing	Total
English	420	180	600

Table 1
Dataset Stats in IROSTEREO task at PAN 2022.

Model	Accuracy	Precision	Recall	F1-Score	Train	Test
GCN	0.874 ± 0.032	0.868 ± 0.031	0.874 ± 0.033	0.869 ± 0.031	336	84
GAN	0.864 ± 0.033	0.862 ± 0.028	0.868 ± 0.040	0.863 ± 0.032	336	84
GCN (submission)	**0.900**	-	-	-	420	180

Table 2
Evaluation results for IROSTEREO based on 5-fold cross-validation and final result on test set. Each metric is averaged. For fair evaluation, each fold samples in both models are identical.

4.2. Metrics

Commonly used performance measures include accuracy, precision, recall, and F-measure. However, due to the less number of training data, we mainly focused on the 5-fold cross-validation technique in experimental analysis and model selection phases. In this task, because the data is balanced, accuracy is the main metric.

4.3. Training Setups

The preprocessing is applied to the text, first. Next, using the train set and cross-validation technique we made hyperparameters tuning. We also experimented with Graph Attention Networks (GANs) [20] for model selection between GCN and GAN. In the end, both models worked similarly, however, we chose GCN because of less complexity regarding GANs. The evaluation metrics for each model (GAN and GCN) on cross-validation have been reported in the results section.

5. Results

We experimented using a 5-fold cross-validation methodology to analyze GNNs. Table 2 shows the experimentally designed GCN and GAN models. In the following, we discussed the main quantitative findings and the quantitative analysis of obtained models.

5.1. Main Quantitative Findings

We made experimentation with both GAN and GCN models. According to this experimentation and the results in the table 2, in which the models were trained and evaluated with similar hyperparameters and identical datasets at each fold at the time of 5-fold cross-validation, the main quantitive findings are:

- A comparison of 5-fold cross-validations shows that there is a minor difference between GCN and GAN models so both models behave similarly.

- Both models predictions over the test set revealed the 96% of agreements between models on predictions, again it confirms that both models behave similarly in 96% of the time.
- In model selection, first, due to the high RAM usage of the GAN model, and second, due to small differences between GAN and GCN (which shows GCN is 0.01 better than GAN), we decided to submit the GCN model as a final model for the task.
- In a final submission, we obtained an accuracy of **0.900** over an unseen test set. It again confirms the quality proposed model and word-association importance in capturing dependency information via GNNs.

5.2. Quantitative Analysis

According to our hypothesis, we believe finding word associations allow us to capture the author's style where the primary intent is to spread stereotypes in the form of irony. The proposed method is capable of using heterogeneous graphs to identify irony and stereotypes. In this analysis, we point out the possible reason behind the effectiveness of our approach according to the history of AP task and text-based GCN model advantages in tackling this kind of problem.

Sensitivity to the amount of the data: Deep learning-based models are data-hungry as a result of this deep learning models don't perform on a low number of data for classifications. In previous AP shared tasks only [21] was able to be in the top performer team that used ensemble features to feed fully connected neural networks that allowed them to properly identify the fake news spreaders. This year there was a slight increase in the number of samples however still deep learning models require more amount of data. Because of this the limited data leads to low-quality features for models to capture the author's styles unless bringing more features as a result of these high-level feature extractors is crucial to obtain better performance where a similar story has been investigated in [22] where participants used convolutional neural network (CNNs) to extract valuable features for a classifier to detect hate-speech spreaders. As a result of limited data for irony and stereotypes, it is hard to identify the author's styles. However, according to [17] the Text GCN can achieve higher test accuracy with limited labeled samples. The reason behind this is that similar to CNN models the GCN tries to capture high-level features using word associations in heterogeneous text graphs which allows the classifier to detect the author's styles when they are intended to spread irony and stereotypes.

Effects of the size of authors tweets: In classification problems when there are short texts, the recurrent neural networks (RNNs) specially long-short term memory (LSTM) are performing well. However, the challenge rasses when there are long texts. It is difficult to represent them and also a challenge for RNNs to extract sequential information appropriately. Word order matters for tasks such as sentiment analysis so RNN models are the right choice for these tasks. However, in this task, the word co-occurrences together with word semantics are important, because the aim is to capture styles and author intents. Each author may have its word category to use and by capturing these word preferences we may come across preferable word usages for different classes. With GCNs we can employ word co-occurrences and relations between words in form of edge weights in addition to word semantics in form of nodes. So, GCN doesn't care about word orders and it only cares about word usage with semantics and which makes GCNs a good choice for this task.

Visualization: The figure 2 represents the visualization of the GCN layer for train and test sets using the t-SNE technique. We set the irony labels with red color and not irony with green colors. For the train, we have used the grand truth labels and for the test, we have used the prediction labels. Given author embedding learned by GCN, the t-SNE visualization shows that GCN learns more discriminative author embeddings that are more distinguishable. In the test set, even If we consider a 10 percent of error in prediction the visualization shows a very high-quality split. Generally, the visualizations show that GCN produces linear features for the classifier. The features entered into the linear layer followed by a softmax function, which is formally known as a logistic regression classifier. All together, GCN is capable of extracting linear features which is very well-suited for logistic regression in irony and stereotype spreader identifications.

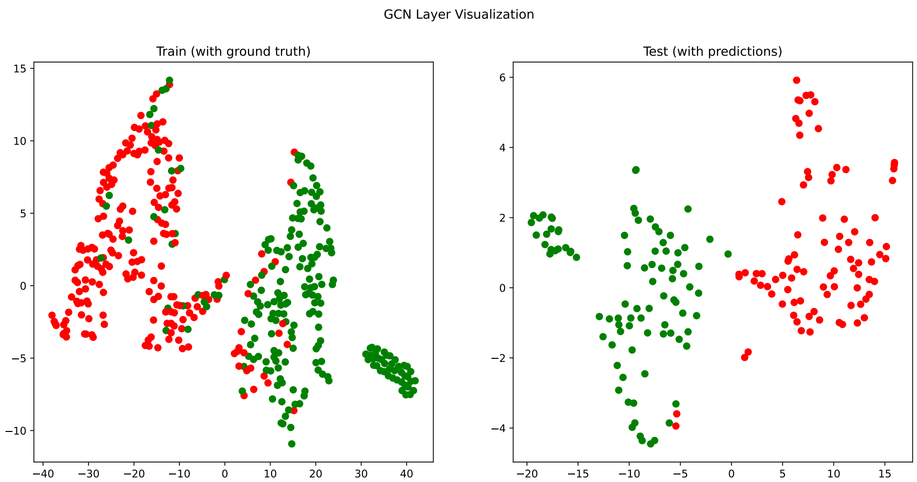

Figure 2: The t-SNE visualization of the GCN layer of author embeddings learned from the AP dataset (red: irony, green: not irony).

6. Conclusion

In this study, we propose a GCN that builds a heterogeneous word authors graph for a whole corpus and turn authors classification into a node classification problem. The GCN can capture global word co-occurrence information and utilize limited labeled documents well. A simple single-layer Text GCN demonstrates promising results for Profiling Irony and Stereotype Spreaders on the Twitter task in PAN 2022. In the final, we achieved an accuracy of 0.900. Based on our manual evaluation, our approach is very capable of determining irony and stereotype spreaders.

References

[1] 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.

[2] 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.

[3] F. Rangel, P. Rosso, Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling, in: L. Cappellato, N. Ferro, D. Losada, H. Müller (Eds.), CLEF 2019 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2019. URL: http://ceur-ws.org/Vol-2380/.

[4] F. Rangel, A. Giachanou, B. Ghanem, P. Rosso, Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/.

[5] F. Rangel, P. Rosso, G. L. D. L. P. Sarracén, E. Fersini, B. Chulvi, Profiling Hate Speech Spreaders on Twitter Task at PAN 2021, in: A. J. M. M. F. P. Guglielmo Faggioli, Nicola Ferro (Ed.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[6] T. Funke, T. Guo, A. Lancic, N. Antulov-Fantulin, Low-dimensional statistical manifold embedding of directed graphs, 2020. `arXiv:1905.10227`.

[7] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE transactions on neural networks 20 (2008) 61–80.

[8] M. Rahgouy, H. Babaei Giglou, T. Rahgooy, C. Seals, Null at SemEval-2022 Task 6: Intended Sarcasm Detection Using Stylistically Fused Contextualized Representation and Deep Learning, in: Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Association for Computational Linguistics, 2022.

[9] H. Giglou, T. Rahgooy, J. R. M. Rahgouy, Z. Rahgooy, Profiling Haters on Twitter using Statistical and Contextualized Embeddings—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-154.pdf.

[10] H. Babaei Giglou, T. Rahgooy, M. Rahgouy, J. Razmara, UoT-UWF-PartAI at SemEval-2021 task 5: Self attention based Bi-GRU with multi-embedding representation for toxicity highlighter, in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Association for Computational Linguistics, Online, 2021, pp. 948–952. URL: https://aclanthology.org/2021.semeval-1.129. doi:`10.18653/v1/2021.semeval-1.129`.

[11] P. Mishra, M. Del Tredici, H. Yannakoudakis, E. Shutova, Abusive Language Detection with Graph Convolutional Networks, 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 Linguis-

tics, Minneapolis, Minnesota, 2019, pp. 2145–2150. URL: https://aclanthology.org/N19-1221. doi:10.18653/v1/N19-1221.
[12] J. S.-J. y Paolo Rosso y Manuel Montes-y-Gómez y Berta Chulvi, Masking and bert-based models for stereotype identication, Procesamiento del Lenguaje Natural 67 (2021) 83–94. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6379.
[13] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, M. Sanguinetti, SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 54–63. URL: https://aclanthology.org/S19-2007. doi:10.18653/v1/S19-2007.
[14] D. I. H. Farías, V. Patti, P. Rosso, Irony detection in twitter: The role of affective content, ACM Trans. Internet Technol. 16 (2016). URL: https://doi.org/10.1145/2930663. doi:10.1145/2930663.
[15] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, CoRR abs/1609.02907 (2016). URL: http://arxiv.org/abs/1609.02907. arXiv:1609.02907.
[16] S. C. Han, Z. Yuan, K. Wang, S. Long, J. Poon, Understanding graph convolutional networks for text classification, 2022. URL: https://arxiv.org/abs/2203.16060. doi:10.48550/ARXIV.2203.16060.
[17] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, CoRR abs/1809.05679 (2018). URL: http://arxiv.org/abs/1809.05679. arXiv:1809.05679.
[18] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, CoRR abs/1908.10084 (2019). URL: http://arxiv.org/abs/1908.10084. arXiv:1908.10084.
[19] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, ArXiv abs/1910.01108 (2019).
[20] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, stat 1050 (2017) 20.
[21] H. Giglou, J. Razmara, M. Rahgouy, M. Sanaei, LSACoNet: A Combination of Lexical and Conceptual Features for Analysis of Fake News Spreaders on Twitter—Notebook for PAN at CLEF 2020, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/.
[22] M. Siino, E. Di Nuovo, I. Tinnirello, M. La Cascia, Detection of hate speech spreaders using convolutional neural networks—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-189.pdf.