Precog-LTRC-IIITH at GermEval 2021: Ensembling Pre-Trained Language Models with Feature Engineering

T. H. Arjun and **Arvindh A.** and **Ponnurangam Kumaraguru**
Precog-LTRC, International Institute of Information Technology, Hyderabad, India
{arjun.thekoot,arvindh.a}@research.iiit.ac.in
pk.guru@iiit.ac.in

Abstract

We describe our participation in all the subtasks of the Germeval 2021 shared task on the identification of Toxic, Engaging, and Fact-Claiming Comments. Our system is an ensemble of state-of-the-art pre-trained models finetuned with carefully engineered features. We show that feature engineering and data augmentation can be helpful when the training data is sparse. We achieve an F1 score of 66.87, 68.93, and 73.91 in Toxic, Engaging, and Fact-Claiming comment identification subtasks.

1 Introduction

Facebook quickly rose in popularity around 2008, taking the world by storm single-handedly creating the initial social media buzz. Its user base is steadily increasing ever since and has held its position as the most used platform ever since the early 2010s.[1] It has around 2.38 billion users, and the increase hasn't flattened yet. The initial purpose of such social media platforms was to establish a bridge for fruitful information exchange, which is currently inhibited by offensive language and misinformation spread. Given the number of comments exchanged each day, it's impossible to manually classify and mitigate such behavior.

GermEval is a series of shared task evaluation campaigns that focus on natural language processing for the German language. GermEval 2021 tasks are intended to classify comments on Facebook into three categories of Toxic, Engaging, and Fact-Claiming comments. Subtask A focuses on the identification of offensive language which could be used to ban/timeout these users. Subtask B on Fact-claiming can further be classified as misinformation, and Subtask C on engaging comments promoting cleaner information exchange. The outline of this paper is as follows: We give a short overview of related work in Section 2. We then describe the dataset provided in Section 3 and the preprocessing techniques we use in Section 4, explain the features we engineered in Section 5, and the architecture of our solution in Section 6. We then move onto the evaluation of our solution in Sections 7-9 and conclude in Section 10.

2 Related Work

2.1 Toxic Comment Classification

There have been various shared tasks and competitions in this task such as: GermEval Task 2, 2019 (Struß et al., 2019), GermEval 2018 (Wiegand et al., 2018), SemEval 2019 - Task 5 (Basile et al., 2019), SemEval 2019 - Task (OffensEval 2019) (Zampieri et al., 2019), SemEval 2020 (Zampieri et al., 2020), Kaggle's Toxic Comment Classification Challenges.[2]

Wu et al. (2019) use the BERT model to detect and classify offensive language in English tweets and obtain good results. Risch and Krestel (2020b) discuss toxic comments in online news discussions and describe subclasses of toxicity, present various deep learning approaches, and propose to augment training data by using transfer learning when the training data is sparse.

2.2 Engaging Comment Classification

Risch and Krestel (2020a) analyze user engagement in the form of the upvotes and replies that the comments receive. They train a model to classify based on text and achieve excellent results with RNN and CNN models. They also analyze what makes each comment engaging. Ambroselli et al. (2018) use a Logistic Regression Model with metadata, along with extracted semantic and linguistic

[1] Statistics https://bit.ly/3AZdQtj

[2] kaggle-challenge https://bit.ly/3hZMYAx

Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments
co-located with KONVENS

features. Napoles et al. (2017) use a CNN with word embeddings to classify engaging threads.

2.3 Fact-Claiming Comment Classification

Chatterjee et al. (2018) propose combining BOW and manually engineered features for classifying facts and opinions on Twitter and show that handcrafted textual features could help in the task. Hassan et al. (2015) propose a feature-based method in which sentiment, TF-IDF, part-of-speech, and other descriptive features are fed into classical models, such as SVMs. There have been various other deep learning-based attempts as well (Atanasova et al., 2018). Meng et al. (2020) identify fact-claiming text using a Bert Model and use adversarial training to avoid overfitting.

All previous attempts at these tasks show how feature engineering and deep learning approaches can be helpful in these tasks.

3 Dataset

The dataset provided for the shared task (Risch et al., 2021) is an annotated dataset of Facebook user comments that four trained annotators have labeled. The dataset was collected from the Facebook page of a political talk show of a German television broadcaster (information about which was not revealed to the participants), consisting of user discussions from February till July, 2019. The dataset provided is anonymized. Links to users are replaced by @USER, likewise links to the show replaced by @MEDIUM, and the links to the show's moderator replaced by @MODERATOR. Each comment of the dataset is annotated into three categories - Toxic, Fact-Claiming, and Engaging. The test set contains 944 comments extracted from different shows other than the one in the training data. This way, the participants were provided with a realistic use case and could possibly test a possible bias caused by topics of discussion. There is an imbalance in the distribution of classes in the given dataset. Still, we let the models be biased with this class imbalance as we believe it provides our models a fair understanding of these distributions from the real world.

4 Data Preprocessing

The corpora is abundant in emojis. We transcribe all emojis into German text instead of removing them while cleaning the text as not to lose information present, such as emotions. For this, we use a transliteration mapping for emojis.[3] We use the googletrans library[4] to translate these to German. We remove hyperlinks, mentions, lower-case the text, remove punctuations except for apostrophes and periods, and perform Unicode normalization. Due to the limit on the number of tokens (512 tokens) for transformer-based models, we cut the text in the middle of the sentence i.e. we take the first 500 and last 12 tokens.

5 Feature Engineering

We look at the linguistic features of the text and explore their correlation with each categorical prediction.

5.1 Stylistic Features

Features include total length, number of unique words, words, exclamation, question marks, all capital words, the percentage of unique words, other punctuations, URLs, distribution of emojis, etc.

5.2 Linguistic Features

We use a list of German stopwords from the nltk library (Loper and Bird, 2002) and use their distribution as a feature. We use SentiWS Dataset (Remus et al., 2010), which provides negative and positive sentiment scores for words. We use this to get the percentage of negative and positive sentiment scores. We use Language Tool[5] which can detect a variety of linguistic anomalies, including grammatical errors, missing punctuation, or wrong capitalization. We note these errors and we propose the distribution of them as a feature. FTR Classifier[6] is a natural language classifier that uses keyword-based methods to identify future-referring sentences and whether they use the present tense, future tense, or express epistemic certainty or uncertainty. It also has a list of German past, future, uncertain, certain words which we use. German grammatical features such as Partizip (Participle), Partizip II (Past Participle), Präteritum (Preterite) are taken from the German Verbs Database.[7] We note down the distribution of the 10 verb categories mentioned in the database. We also use the Dale Chall Readability Index Calculation (Dale and Chall, 1948) to find the

[3] Emoji-list https://bit.ly/3wtom8E
[4] googletrans https://bit.ly/3yNW0Y5
[5] language-tool-python https://bit.ly/3yKb6Og
[6] FTR Classifier https://bit.ly/3xER8Vk
[7] German-Verbs-Database https://bit.ly/3i1BDzZ

Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments
co-located with KONVENS

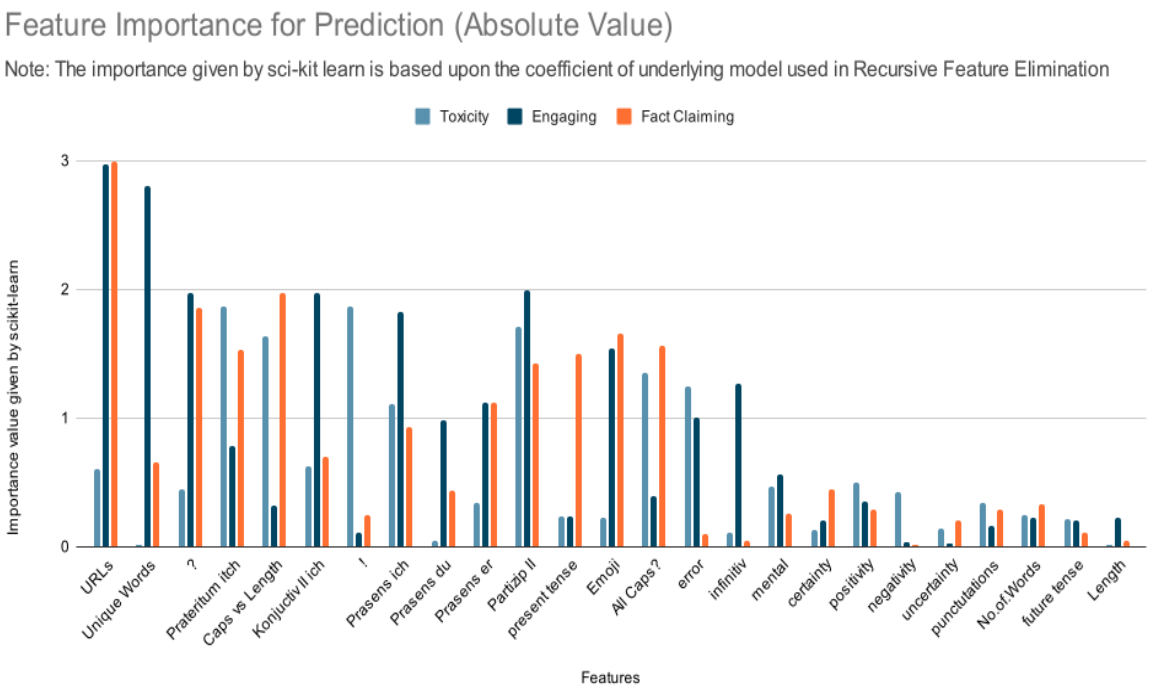

Figure 1: We look at the importance of features using scikit-learn feature selection and choose 20 important features. The plot shows the importance marked by feature selection (absolute values).

readability of the text. For applying this to the German text, we use the python library readability.[8]

We perform feature selection on these handcrafted features using two filters, Pearson correlation and scikit-learn library's[9] feature selection. We choose the top 20 that we presume to be essential for our models.

Figure 1 shows the importance (absolute values) marked by the scikit-learn library's feature selection. After feature selection, we select the following 20 features:

- Readability
- Number of '!', '?', words, URLs
- Percentage of all Capital Words, Partizip II, Präteritum, Punctuations, Linguistic Errors, Präsens ich, words in present, and future tense, unique words, "certainty" and "uncertainty" words.
- Positive and Negative Sentiment score
- Moderator mentions
- Distribution of emojis

[8]readability https://bit.ly/2U1XKym
[9]scikit-learn https://bit.ly/3i6j8ut

6 Model Architectures

We formulate the tasks as a Multi-Label Classification Problem as we are trying to address all 3 tasks. To learn the correlation between these classes, all our models are three-headed which output probabilities for 3 classes corresponding to each subtask. Devlin et al. (2019) achieve the best performance when they concatenate the last 4 hidden layers of the pre-trained network for sentence-level tasks, so in all our models, we use the same approach and concatenate the last 4 hidden layers. We experiment with the following models and techniques (after freezing the pre-trained weights):

6.1 Models

- **Pretrained Transformer Based Models with CNN head**: In this approach, we freeze the pre-trained layers and pass the embedding (concatenated last 4 hidden layers) to a CNN. Kim (2014) report state-of-the-art performance on sentence-level classification after max-pooling convolution layers of various widths to a fully connected layer with dropout. We follow a similar approach where we pass the concatenated last 4 hidden layers of the pre-trained model to convolution lay-

ers of filter sizes 2,3,4,5, on which we apply Max Pooling of pool size 3. We concatenate these outputs with a dropout of 0.5, which is then passed onto a Dense Layer of size 128 with ReLu activation succeeded by a dropout of 0.5. We concatenate this 128 dimensional vector with our 20 dimensional hand-crafted feature vector. We pass this output onto a dense output layer of dimension 3 with sigmoid activation.

- **Pretrained Transformer Based Models with Capsule Net head**: In the image classification domain, capsule networks (Hinton et al., 2011; Sabour et al., 2017) prove to be effective at understanding spatial relationships. Kim et al. (2018) apply this network structure to the classification of text and show its advantage. They argue that CNNs could extract features, but CNNs cannot understand the spatial and proportional relationships between objects in the images or words. Capsule networks address this problem by learning the spatial relationships between words (in text) using additional encoded information. We apply this network architecture with pre-trained embeddings. We pass the pre-trained embeddings through a Bi-Directional GRU Layer of dimension 128 with ReLu activation and dropout of 0.25. We pass this through a Capsule Network of 5 Capsules, 4 routings, and squash activation. This is followed by a dropout of 0.25 and concatenation with our hand-crafted feature vector. This is then passed onto a 3-dimensional dense output layer with sigmoid activation.

- **Fastext and Glove Embeddings with RNN-GRU head**: Along with the transformer-based models, we train word embedding-based models with a RNN head. Unlike transformer-based models, which use sub-word tokenization, the word embedding models could face Out Of Vocabulary (OOV) words. Therefore, we add an extra data cleaning step to reduce the number of OOV words. We deploy a spell checker and correct spellings if possible. For the embeddings layer, we concatenate German fastText (Grave et al., 2018) and German Glove Embeddings (Pennington et al., 2014).[10] We then pass the embeddings through a dropout of 0.5 followed by Bi-Directional LSTM of kernel size 40. This is then passed through a Bi-Directional GRU of the same kernel size. We concatenate the average pool, maximum pool, and the last layer output with our hand-crafted feature vector. This is then passed onto a dense output layer of size 3 with sigmoid activation.

6.2 Ensembling

Our approach uses two levels of Ensembling:

- **Fold Level Ensembling**: We implement early stopping and save the best checkpoint during k-Fold validation for each proposed model. We make a prediction on the test set for each best checkpoint, which we average out to get the best prediction over the k-folds.

- **Model-Level Ensembling**: The predictions of each of the proposed models for each of the pre-trained language models are averaged.

7 Experiments

7.1 Training Data Augmentation

Since the training data is sparse, we follow the approach by Risch and Krestel (2018) where we augment the training set by translating the text to English and then back again to German. We reuse googletrans library for this. This can give us different forms of the same text. Thanks to the accuracy of Google Translate and assuming the meaning remains the same, we can also assume that the labels remain the same. We randomly pick 600 comments for training from this augmented dataset and concatenate them with the given training set.

The models output probabilities for each class. When the value of an output unit is above a given threshold, the corresponding label is predicted. The optimum was found by varying the threshold for the validation set during k-Fold Validation.

7.2 Baseline

We train a Bert finetuned baseline to compare our models against. The Bert model is finetuned for 7 epochs with early stopping and 10-Fold Cross-Validation. This has a classification head on top of the concatenated last 4 hidden layer CLS Token for sentence classification. We consider this a solid

[10] German glove embeddings by deepset.ai https://bit.ly/3xwE58a

Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments
co-located with KONVENS

baseline as it is an ensemble across 10-Fold Cross-Validation of the state-of-the-art Pretrained Language Model, which has proven to be very strong in most cases.

7.3 Experimental Setting

All the above approaches were run on four pre-trained models from huggingface hub[11] namely electra-base-german-uncased[12], German convbert[13], bert-base-german-uncased[14], and a multilingual model xlm-roberta-large.[15] We train the models on each of these pre-trained embeddings and we average the predictions of these models resulting in an ensemble.

We train the models with 10-Fold cross-validation. We use Adam optimizer (Loshchilov and Hutter, 2019) with a learning rate of 1e-3 and a batch size of 32. We train the model for 20 epochs with early stopping with a patience of 3. We didn't experiment with the hyperparameters. The models were implemented using Tensorflow,[16] Keras,[17] and Huggingface Transformers Library.[18] We train on the given dataset with augmentation.

8 Results

Experimenting with the models, we achieve the best performance with an ensemble of models mentioned in the architecture section, which is also our submission. The participants were provided with the gold labels for the test set to evaluate the models. In Table 1, we compare our models on the gold labels and with the baseline model. It is also worth noting that we submitted two system runs. The first one was ensemble of all the individual models listed in Table 1, except models with Capsule Net head. In the second system run, we incorporated models with Capsule Net head into the ensemble (ensemble of all individual models in Table 1). The second system run performed better; hence we centered the analysis around it.

9 Analysis

We carry out an analysis of the test set gold labels to find where our models failed. We find that many misclassified comments were very long ones with more than 512 tokens truncated in the middle part. We truncated in the middle as most of the emotions must be concentrated at the two ends. A possible solution could be is to use hierarchical LSTMs with chunking of 512 token chunks of these texts and feeding them to the models or using longformer based models (Beltagy et al., 2020). We analyze some of the misclassified texts by our model below. They were translated by a native German, two non-native speakers, and google translate. (Note: The translations given below are the ones by the native speaker)

1. "Großen Respekt wie Herr Hallervorden mit der Situation und seinen Mitarbeitern umgeht. Wenn es nach Herrn Lauterbach gehen würde ,würden sie es im stillen Kämmerlein aussitzen."
   translates to
   "I pay a lot of respect to how Mr. Hallervorden is dealing with the situation and his co-workers. If it were up to Mr. Lauterbach, they would keep it under the table."

2. "@USER Wissen sie was oder reden Sie einfach auch völlig unfundiert daher? Wenn sie was wissen lassen sie uns an ihrem Wissen teilhaben!"
   translates to
   "@USER Do you know something or are you also speaking fully in unfounded terms? If you know something, let us know about the knowledge you have!"

3. "@USER weil er es kann."
   translates to
   "@USER because he can."

4. "@USER dem kann ich nur zustimmen. Was nützt dem Klima eine CO2 Bepreisung? Finde den Fehler. Aber so generiert man unter dem Deckmantel Klimaschutz neue Abgaben, wir alle werden noch mehr zahlen müssen ohne das sich etwas ändert. Bewährtes Verfahren. Immer mit dem Finger auf die anderen zeigen ist ja so einfach"
   translates to
   "@USER I can agree with that. How could a CO2-tax be useful for climate? Find the mistake. But with that you can implement new taxes under the disguise of climate protection.

[11] huggingface-hub https://huggingface.co/models
[12] electra https://bit.ly/3e8zX6w
[13] convbert https://bit.ly/3wB6Qzt
[14] bert https://bit.ly/3yQXqB8
[15] xlm-roberta https://bit.ly/2TUkzEh
[16] tensorflow https://tensorflow.org/
[17] keras https://keras.io/
[18] Huggingface https://huggingface.co/

	SubTask A			SubTask B			SubTask C		
Model	T1 F1	T1 P	T2 R	T2 F1	T2 P	T2 R	T3 F1	T3 P	T3 R
Baseline	**59.38**	**60.54**	**58.27**	**65.27**	**65.92**	**64.64**	**67.19**	**67.47**	**66.9**
FastText Glove RNN	63.59	67.18	60.37	68.71	68.86	68.67	70.24	71.57	68.97
Bert CNN	62.76	65.67	60.10	67.09	69.13	65.17	73.69	76.19	71.35
Bert BertCapsule Net	64.56	66.28	62.93	67.18	68.28	66.13	73.99	75.41	72.63
Electra CNN	64.82	66.56	63.16	67.00	68.37	65.69	73.49	74.20	72.69
Electra Capsule Net	67.80	72.99	63.31	66.52	66.96	66.07	72.72	72.89	72.55
ConvBert CNN	58.94	60.72	57.27	66.06	67.17	64.99	70.32	71.74	68.97
ConvBert Capsule Net	64.17	67.70	61.00	67.28	69.30	65.37	71.94	73.56	70.40
XLM-Roberta CNN	62.01	68.01	56.98	67.65	68.75	66.59	71.87	72.90	70.88
XLM-Roberta Capsule Net	65.05	67.25	63.00	70.26	70.93	69.60	73.95	76.48	71.59
Ensemble Submission	**66.87**	**67.42**	**66.33**	**68.93**	**68.37**	**69.50**	**73.91**	**73.44**	**74.39**

Table 1: Comparison of various models, including baseline across the three tasks in which the ensemble submission incorporates Capsule Net.

> We all will have to pay more without any improvement. Best practice. It is always easy to point finger at others."

In Comment 1, the gold label is toxic. Without context, it could also be classified as non-toxic, since it is congratulatory in the first part. Comments 2 and 3 were classified as toxic but are non-toxic. One could note that both are in a rude tone. This could be because of the context of the comment and to what it is referring to.

For engaging comments, some misclassified comments in our analysis were both toxic and engaging, which is strange without context. For subtask 3, comment 4 was classified as Fact-claiming by the model, but the comment seems to be claiming a practice.

We find in our testing that hand-crafted features could be crucial in improving the performance of pre-trained finetuning for low-resource tasks. We also notice no discrepancy between precision and recall even though there was a class imbalance in the training set. Hence, our hypothesis that the model benefits from learning the class distributions and their correlations in the real world is validated.

10 Conclusion

Participating in all three shared tasks, we submit predictions from a model ensemble. We perform feature engineering and dataset augmentation and show how this can help train neural networks in low-resource tasks. Our model ensemble with hand-crafted features performs better than the baseline Fine-Tuned Bert Model. We also analyze the errors made by our model against the gold label to understand the flaws in the model. We have also made the source code public[19] for reference.

11 Acknowledgments

We want to thank Google Colab for free access to GPUs and TPUs for quick and small experimentation in teams and visualization help, IIIT-H for providing us with access to HPC Cluster with GPUs for training our deep learning models and also to our colleagues at Precog[20] for constant support and feedback.

References

Carl Ambroselli, Julian Risch, Ralf Krestel, and Andreas Loos. 2018. Prediction for the newsroom: Which articles will get the most comments? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 193–199, New Orleans - Louisiana. Association for Computational Linguistics.

[19] Source code https://bit.ly/36zOjJg
[20] Precog Lab https://precog.iiitd.edu.in/

Pepa Atanasova, Alberto Barron-Cedeno, Tamer Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Da San Martino, and Preslav Nakov. 2018. Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims. task 1: Check-worthiness.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. 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*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Swayambhu Chatterjee, Shuyuan Deng, Jun Liu, Ronghua Shan, and Wu Jiao. 2018. Classifying facts and opinions in twitter messages: a deep learning-based approach. *Journal of Business Analytics*, 1(1):29–39.

Edgar Dale and Jeanne S. Chall. 1948. A formula for predicting readability. *Educational Research Bulletin*, 27(1):11–28.

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)*, pages 4171–4186. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Naeemul Hassan, Chengkai Li, and Mark Tremayne. 2015. Detecting check-worthy factual claims in presidential debates. *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1835–1838.

Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. 2011. Transforming auto-encoders. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I*, page 44–51. Springer-Verlag.

Jaeyoung Kim, Sion Jang, Sungchul Choi, and Eunjeong Park. 2018. Text classification using capsules.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Kevin Meng, Damian Jimenez, Fatma Arslan, Jacob Daniel Devasier, Daniel Obembe, and Chengkai Li. 2020. Gradient-based adversarial training on transformer networks for detecting check-worthy factual claims.

Courtney Napoles, Aasish Pappu, and Joel R. Tetreault. 2017. Automatically identifying good conversations online (yes, they do exist!). In *ICWSM*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. SentiWS - a publicly available German-language resource for sentiment analysis. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).

Julian Risch and Ralf Krestel. 2018. Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 150–158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Julian Risch and Ralf Krestel. 2020a. Top comment or flop comment? predicting and explaining user engagement in online news discussions. In *Proceedings of the International Conference on Web and Social Media (ICWSM)*, pages 579–589.

Julian Risch and Ralf Krestel. 2020b. *Toxic Comment Detection in Online Discussions*, pages 85–109.

Julian Risch, Anke Stoll, Lena Wilms, and Michael Wiegand. 2021. Overview of the GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments co-located with KONVENS*, pages 1–12.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules.

Julia Struß, Melanie Siegel, Josef Ruppenhofer, Michael Wiegand, and Manfred Klenner. 2019. Overview of germeval task 2, 2019 shared task on the identification of offensive language.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language.

Zhenghao Wu, Hao Zheng, Jianming Wang, Weifeng Su, and Jefferson Fong. 2019. BNU-HKBU UIC NLP team 2 at SemEval-2019 task 6: Detecting offensive language using BERT model. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 551–555, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447. International Committee for Computational Linguistics.