NLytics at CheckThat! 2022: Hierarchical multi-class fake news detection of news articles exploiting the topic structure

Albert Pritzkau[1], Olivier Blanc[2], Michaela Geierhos[2] and Ulrich Schade[1]

[1]*Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE, Fraunhoferstraße 20, 53343 Wachtberg, Germany*

[2]*Research Institute Cyber Defence (CODE), University of the Bundeswehr Munich, Carl-Wery-Straße 18, 81739 München, Germany*

Abstract

The following system description presents our approach to the detection of fake news in texts. The given task has been framed as a multi-class classification problem. In a multi-class classification problem, each input chunk is assigned one of several class labels.

To dissect content patterns in the training data, we made use of topic modeling. Topic modeling techniques such as Latent Dirichlet Allocation (LDA) are unsupervised algorithms that pick up on patterns and provide an estimate of what the messages convey.

In order to assign class labels to the given documents, we opted for RoBERTa (A Robustly Optimized BERT Pretraining Approach) and Longformer as neural network architectures for sequence classification. Starting off with a pre-trained model for language representation, we fine-tuned this model on the given classification task with the provided annotated data in supervised training steps. In a hierarchical approach, the training of a classifier took place at topic level.

Keywords

Sequence Classification, Deep Learning, Transformers, RoBERTa, Longformer, Topic modeling

1. Introduction

The proliferation of disinformation online has given rise to a lot of research on automatic fake news detection. CLEF - CheckThat! Lab [1, 2] considers disinformation as a communication phenomenon. By detecting the use of various linguistic features in communication, the given task takes into account not only the content but also how a subject matter is communicated.

The Shared Task 3 of the CLEF 2022 - CheckThat! Lab[3] defines the following subtasks:

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

albert.pritzkau@fkie.fraunhofer.de (A. Pritzkau); olivier.blanc@unibw.de (O. Blanc); michaela.geierhos@unibw.de (M. Geierhos); ulrich.schade@fkie.fraunhofer.de (U. Schade)

https://www.fkie.fraunhofer.de (A. Pritzkau); https://www.unibw.de/code (O. Blanc); https://www.unibw.de/code (M. Geierhos); https://www.fkie.fraunhofer.de (U. Schade)

0000-0001-7985-0822 (A. Pritzkau); 0000-0002-2546-857X (U. Schade)

Subtask 3A Given the textual content of an article, specify a credibility level for the content ranging between “true”, “false”, “partially false” including “other”.

Subtask 3B Transfer learning task to build a classification model for the German language along with the previous multi-class task.

This paper covers our approach on the multi-class classification task to detecting fake news. To build our models, only textual content is given as input. Below, we describe the system built for subtask 3A. At the core of our systems pre-trained models based on the Transformer architecture [4] such as RoBERTa [5] or Longformer [6] were used.

2. Related Work

The goal of the shared task is to investigate automatic techniques for identifying various rhetorical and psychological features of disinformation campaigns. A comprehensive survey on fake news and on automatic fake news detection has been presented by Zhou and Zafarani [7]. Based on the structure of data reflecting different aspects of communication, they identified four different perspectives on fake news: (1) the false knowledge it carries, (2) its writing style, (3) its propagation patterns, and (4) the credibility of its creators and spreaders.

CLEF 2022 CheckThat! Lab Task 3 emphasizes communicative styles that systematically co-occur with persuasive intentions of (political) media actors. Similar to de Vreese et al. [8], propaganda and persuasion is considered as an expression of political communication content and style. Hence, beyond the actual subject of communication, the way it is communicated is gaining importance[9].

We build our work on top of this foundation by first investigating content-based approaches for information discovery. Traditional information discovery methods are based on content: documents, terms, and the relationships between them [10]. The methods can be considered as general Information Extraction (IE) methods, automatically deriving structured information from unstructured and/or semi-structured machine-readable documents. Communities of researchers have contributed various techniques from machine learning, information retrieval, and computational linguistics to the different aspects of the information extraction problem. From a computer science perspective, existing approaches can be roughly divided into the following categories: rule-based, supervised, and semi-supervised. In our case, we followed the supervised approach by reframing the complex language understanding task as a simple classification problem. Text classification, also known as text tagging or text categorization, is the process of categorizing text into organized groups. By using Natural Language Processing (NLP), text classifiers can automatically analyze human language texts and then assign a set of predefined tags or categories. Historically, the evolution of text classifiers can be divided into three stages: (1) simple lexicon- or keyword-based classifiers, (2) classifiers using distributed semantics, and (3) deep learning classifiers with advanced linguistic features.

2.1. Deep Learning and Pre-trained Deep Language Representation

Recent work on text classification uses neural networks, particularly Deep Learning (DL). Badjatiya et al. [11] demonstrated that these architectures, including variants of Recurrent Neural Networks (RNN) [12, 13, 14], Convolution Neural Networks (CNN) [15], or their combination (CharCNN, WordCNN, and HybridCNN), produce state-of-the-art results and outperform baseline methods (character n-grams, TF-IDF or bag-of-words representations).

Until recently, the dominant paradigm in approaching NLP tasks has been focused on the design of neural architectures, using only task-specific data and word embeddings such as those mentioned above. This led to the development of models such as Long Short Term Memory (LSTM) networks or Convolution Neural Networks, which achieve significantly better results in a range of NLP tasks as compared to less complex classifiers such as Support Vector Machines, Logistic Regression or Decision Tree Models. Badjatiya et al. [11] demonstrated that these approaches outperform models based on character and word n-gram representations. In the same paradigm of pre-trained models, methods like BERT [16] and XLNet [17] have been shown to achieve state-of-the-art performance in a variety of tasks.

Indeed, the usage of a pre-trained word embedding layer to convert the text into vectorized input for a neural network marked a significant step forward in text classification. The potential of pre-trained language models, e.g. Word2Vec [18], GloVe [19], fastText [20], or ELMo [21], to capture the local patterns of features to benefit text classification, has been described by Castelle [22]. Modern pre-trained language models use unsupervised learning techniques such as creating RNN embeddings on large text corpora to gain some primal "knowledge" of the language structures before a more specific supervised training steps in. Transformer-based models are unable to process long sequences due to their self-attention mechanism, which scales quadratically with the sequence length. BERT-based models enforce a hard limit of 512 tokens, which is usually enough to process the majority of sequences in most benchmark datasets.

2.2. BERT, RoBERTa and Longformer

BERT stands for Bidirectional Encoder Representations from Transformers. It is based on the Transformer model architectures introduced by Vaswani et al. [4]. The general approach consists of two stages: first, BERT is pre-trained on vast amounts of text, with an unsupervised objective of masked language modeling and next-sentence prediction. Second, this pre-trained network is then fine-tuned on task specific, labeled data. The Transformer architecture is composed of two parts, an Encoder and a Decoder, for each of the two stages. The Encoder used in BERT is an attention-based architecture for NLP. It works by performing a small, constant number of steps. In each step, it applies an attention mechanism to understand relationships between all words in a sentence, regardless of their respective position. By pre-training language representations, the Encoder yields models that can either be used to extract high quality language features from text data, or fine-tune these models on specific NLP tasks (classification, entity recognition, question answering, etc.). We rely on RoBERTa [5], a pre-trained Encoder model which builds on BERT's language masking strategy. However, it modifies key hyperparameters in BERT such as removing BERT's next-sentence pre-training objective, and training with much larger mini-batches and learning rates. Furthermore, in comparison to BERT, the training data set for

Roberta was an order of magnitude larger (160 GB of text) with the maximum sequence length of 512 used for all interations. This allows RoBERTa representations to generalize even better to downstream tasks.

To address the limitation of traditional Transformer-based models to 512 tokens, Longformer[6] uses an attention pattern that scales linearly with sequence length, making it easy to process documents of thousands of tokens or longer. To this end, the standard self-attention is replaced by an attention mechanism, which combines a local windowed attention with a task motivated global attention, thus allowing up to 4096 position embeddings. Longformer is pre-trained from RoBERTa[5].

3. Dataset

The training data for this task was developed during the CLEF-2021 CheckThat! campaign [23, 24, 25] and provided by Shahi et al. [26]. The AMUSED framework presented by Shahi [27] was used for data collection. The test data was gathered during CLEF 2022 CheckThat! Lab [2]. The adopted task was framed as multi-class classification problem. Class labels were provided as credibility levels {false, partially false, true, other} as proposed by Shahi et al. [28]. The provided training set consists of 1,264 documents. As suggested by the organizers, a much larger training set was collected, combining data sets from comparable tasks such as the Fake News Detection Challenge KDD 2020 [29], as well as the Fake News Classification Datasets [30]. The resulting large training corpus also mentioned in [31] consists of 51148 documents.

Table 1
Composition of corpora used for training

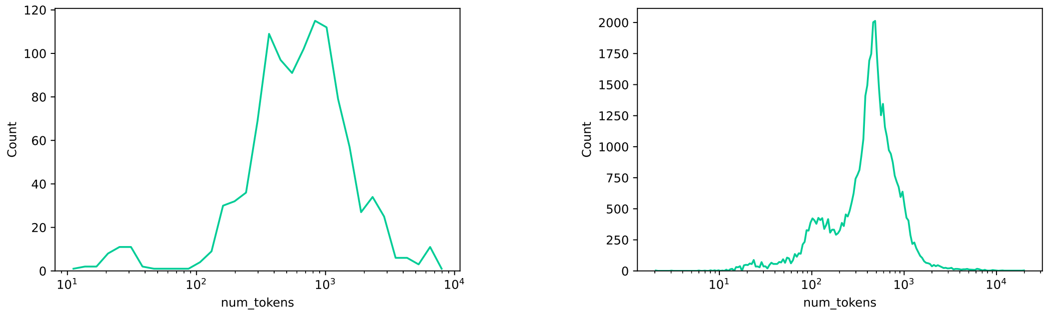
large training corpus	original training data	1264
	Fake News Detection Challenge KDD 2020	4986
	Fake News Classification Datasets	44898
		51148

The content parts are distributed between title and body of messages. Both fields were concatenated to serve as the input for training.

4. Exploratory data analysis

Our approach is based on a comprehensive exploratory analysis of the training data.

Cleaning The initial training dataset consisted of 1264 documents. The explorative analysis started with the investigation of inconsistencies in the dataset. Unexpectedly, ambiguities in the annotation of the documents could be detected. For example, identical documents were found with contradictory annotations "true" vs. "false". In this case, we decided to remove all affected documents from the training data, regardless of the provided annotation. Removing



(a) original training data

(b) large training corpus

Figure 1: Document length (token-based) distribution in the training sets.

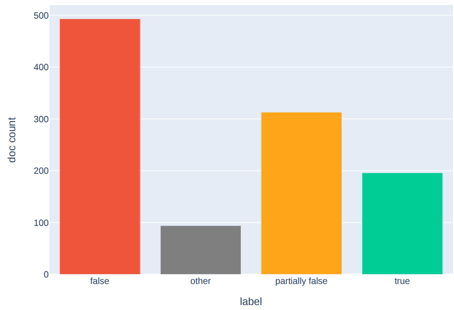
Table 2
Statistical summaries of token (word) counts on all utilized datasets.

	original training data (cleansed)	large training corpus (cleansed)	test data
doc count	1096	44910	612
mean	887.82	521.97	1184.60
std	926.16	638.90	2005.33
min	10	2	60
25%	360.75	261.00	432.50
50%	639.00	442.00	723.00
75%	1065.25	623.00	1179.25
max	8751	20304	22168

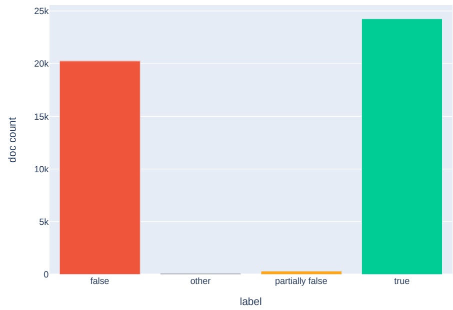
just one of the duplicates would have led to an inadvertent weighting of the remaining class. After the elimination of the ambiguities, remaining unique duplicates could be easily removed. The final cleansed dataset contained 1096 documents. We applied the same procedure to the 44910 documents for the large training corpus. The remainder of this study focuses on this adjusted version of the originbal dataset.

Generally, duplicate data does not add any value since looking at the same data multiple times does not make the algorithm any better. However, if the distribution of duplicates is skewed towards one class only, a bias is to be expected in the resulting classification, throwing off the generalization performance, as the model is given information that overrepresents that class.

Token count The statistical summary of token counts in Table 2 as well as Figure 1 suggests that most of the sequences of the training set exceed the limitation of trditional Transformer-based models to 512 tokens as described previously. Thus, anything beyond this limitation will be truncated. For this reason, after an initial training with RoBERTa at its core, we switched to Longformer[6] as the basic architecture, to gradually improve the overall score.



(a) original training data



(b) large training corpus

Figure 2: Label distribution - training sets

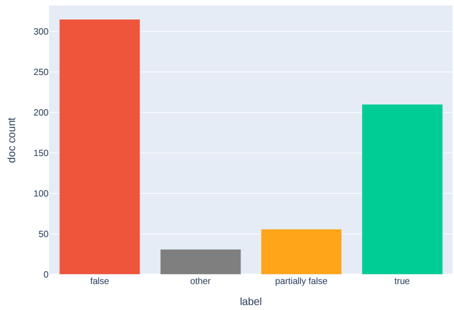


Figure 3: Label distribution in the gold standard.

Unbalanced class distribution Imbalance in data can exert a major impact on the value and meaning of accuracy and on certain other well-known performance metrics of an analytical model. Figure 2 depicts a clear skew towards false information. Furthermore, the "true" class is significantly underrepresented as compared to "partially false" class.

Topic structure To dissect content patterns in the training data, we made use of topic modeling. As unsupervised algorithms, topic modeling techniques such as Latent Dirichlet Allocation[32] (LDA) pick up on patterns and provide an overview of the information that the data contains. To help distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference, topic coherence measures [33] are utilzed measuring the degree of semantic similarity between high scoring words in a given topic. In particular, a series of sensitivity tests were performed (see Figure 4) to help determine the optimal number of topics as an essential model hyperparameter. Throughout the sensitivity tests CV was applied as coherence measure. CV creates content vectors of words using their co-occurrences. It is based on a sliding window, a one-set segmentation of the top words and an indirect confirmation measure that uses normalized pointwise mutual information (NPMI) and the cosinus similarity. To further improve the interpretability of the esulting topics, other coherence measures such as the UMass Coherence Score may also be explored. Based on these tests 15 was chosen as the optimal number of topics, since the coherence score does not change significantly even for a higher number of topics.

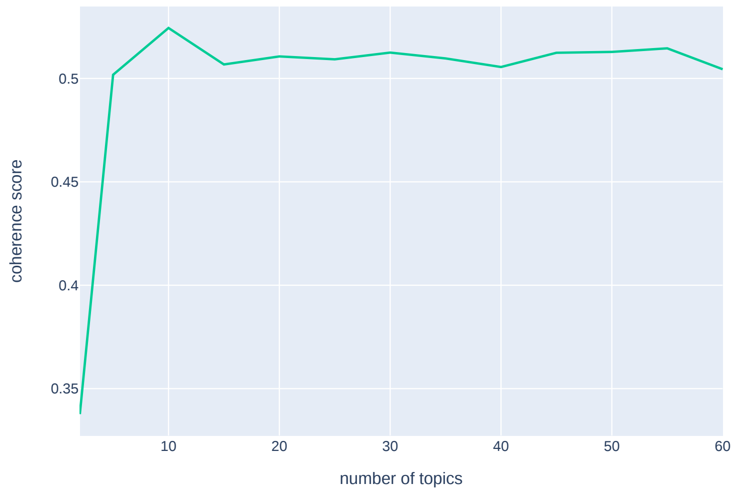


Figure 4: Coherence scores to choose the optimal number of topics.

The resulting topic distributions as well as their high scoring words are depicted in Figure 5. In fact, the distribution of labels differs significantly depending on the topic as shown in Figure 6.

5. Our approach

Our approach is based on the assumption that differentiation of various viewpoints usually takes place in a topic-related manner. A topic results from a specific distribution over the words used. Via this distribution, different topics can be distinguished from each other. With our approach we propose a hierarchical method, where automatic text classification takes place on topic level.

5.1. Experimental setup

Model Architecture Subtask 3A is given as a multi-class classification problem. The models for the experimental setup were based on RoBERTa and Longformer. For the classification task, fine-tuning is initially performed using *RobertaForSequenceClassification*[34] – roberta-base – as the pre-trained model. *RobertaForSequenceClassification* optimizes for a regression loss (Mean-Square Loss) using an AdamW optimizer with an initial learning rate set to 2e-5. Fine-tuning is

Figure 5: Topics in the training data.

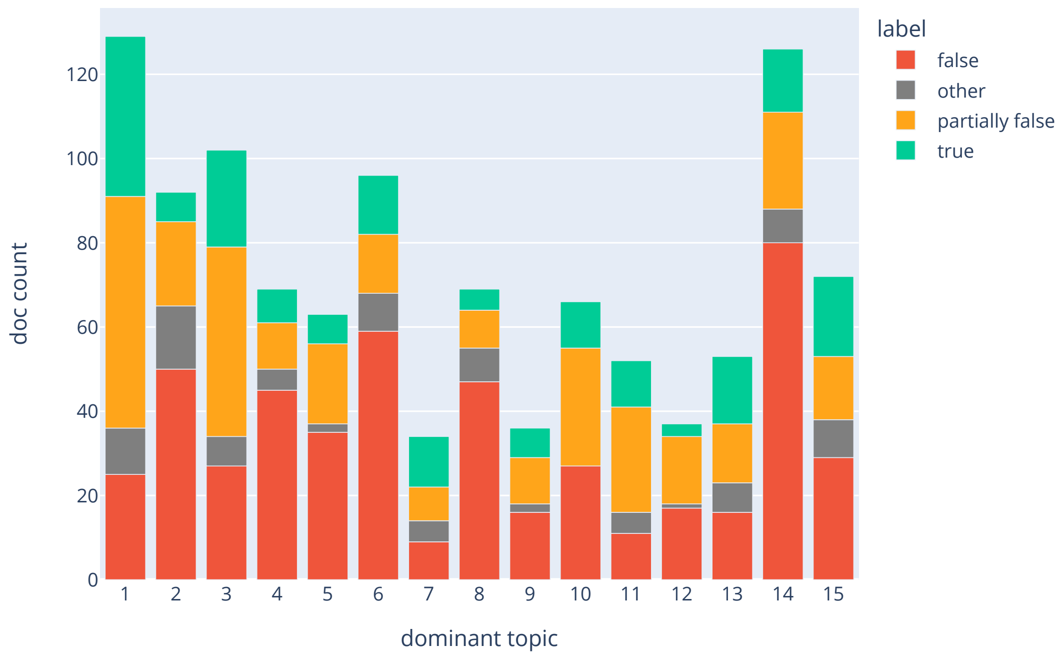


Figure 6: Topic label distribution.

done on NVIDIA TESLA V100 GPU using the Pytorch [35] framework with a vocabulary size of 50265 and an input size of 512. The model is trained to optimize the objective for 10 epochs. To estimate the performance of the resulting models we have chosen a ratio of 82/18 to split the data into training and validation set. We utilized both accuracy and the macro-averaged F1 score to assess the quality of the resulting models. As expected, the RoBERTa model architecture reaches its limitation due to the token counts as shown in Table 2. Therefore, the overall score was significantly improved by replacing the basic architecture with a Longformer configuration which, eventually, was also the architecture utilized for the official submission.

The hierarchical arrangement of text classification is the essential part of our contribution. In this configuration, training and prediction are preceded by topic modeling to first dissect content patterns in the data being mediated. Topics are modelled as distributions over content words derived from documents. To this end, LDA is applied: based on the vocabulary of a document, topics can be assigned to it with a certain probability. The assignment of a particular document to a topic is determined by the highest association probability. The set of documents assigned to a particular topic form the training set for a topic-specific text classifier. Using the model architecture described above, a specific classifier was trained for each derived topic.

Of course, the described hierarchy must also be followed for the model prediction. Based on the previously trained topic model, documents from the test data are first assigned to a topic.

The prediction is then conducted by the dedicated classification model.

Both topic modeling and text classification are implemented in the form of a comprehensive pipeline.

Input Embeddings The input embedding layer converts the inputs into sequences of features: word-level sentence embeddings. These embedding features will be further processed by the subsequent encoding layers.

Word-Level Sentence Embeddings A sentence is split into words $w_1, ..., w_n$ with length of n by the WordPiece tokenizer [36]. The word w_i and its index i (w_i's absolute position in the sentence) are projected to vectors by embedding sub-layers, and then added to the index-aware word embeddings:

$$\hat{w}_i = WordEmbed(w_i)$$

$$\hat{u}_i = IdxEmbed(i)$$

$$h_i = LayerNorm(\hat{w}_i + \hat{u}_i)$$

Target Encoding We encode the target labels using label encoding, although we assume the target variable to be categorical and non-ordinal. Since we do not assume a natural order, the substitution of the respective category by a natural number is done arbitrarily (cf. Table 3). This might pose a challenge and might be replaced by a multi-label binarizer as an analog of the one-hot (or one-of-K) scheme to multiple labels. It might also be useful to investigate the impact of an alternative order of the target encodings on the result.

Table 3
Label encoding map

label	encoding
true	0
false	1
partially false	2
other	3

5.2. Results and Discussion

We participated in subtask 3A. Official evaluation results of the final submission on the test set are presented in Table 7. The entire classification report on this submission in shown in Table 4. Furthermore, the gold standard also allows the derivation of a corresponding confusion matrix (see Figure 7).

We focused on suitable combinations of deep learning methods as well as their hyperparameter settings. Fine-tuning pre-trained language models like RoBERTa or Longformer on downstream tasks has become ubiquitous in NLP research and applied NLP. Even without extensive pre-processing of the training data, we already achieve competitive results and our models can serve

Table 4
Classification report for the final submission against the gold standard.

	precision	recall	f1-score	support
false	0.6432	0.7841	0.7067	315
other	0.0	0.0	0.0	31
partially false	0.1267	0.33934	0.1845	56
true	0.6575	0.2286	0.3392	210
accuracy	0.5131	0.5131	0.5131	
macro avg	0.3569	0.3380	**0.3076**	612
weighted avg	0.5683	0.5131	0.4970	612

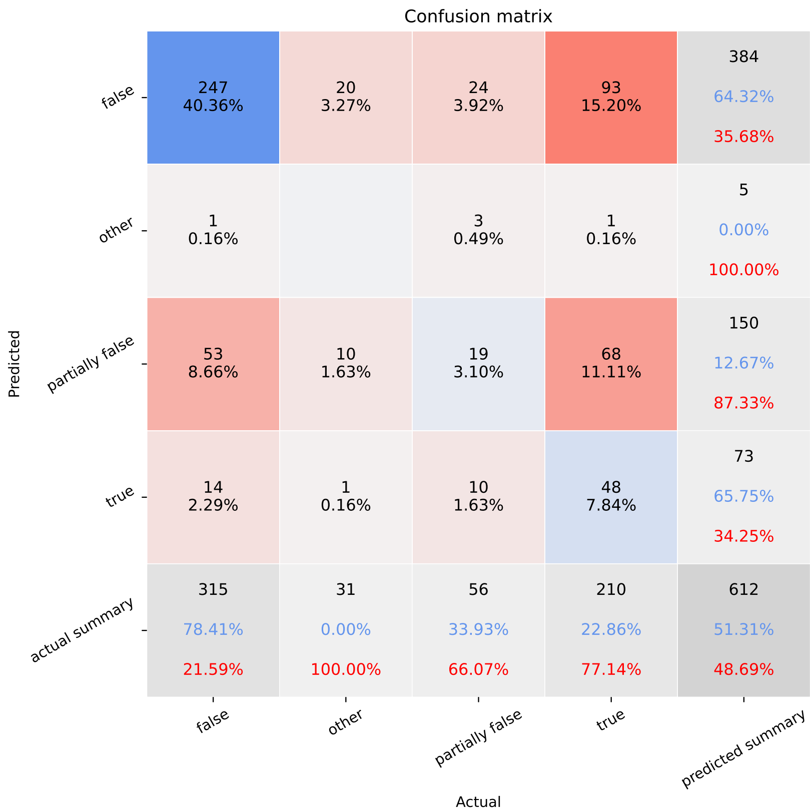


Figure 7: Confusion matrix for Task 3A with the large training corpus on the gold standard.

as strong baseline models which, when fine-tuned, significantly outperform training models trained from scratch. The submission is based on the best performing model checkpoint on the validation set. In our case, of course, this evaluation had to take place at the topic level.

To identify potential improvements, our approach was applied to both the original training dataset and the large training corpus.

When improving on the pretrained baseline models, class imbalance appears to be a primary

Table 5
Classification report for the predictions on the original training data against the gold standard.

	precision	recall	f1-score	support
false	0.5970	0.8889	0.7143	315
other	0.0	0.0	0.0	31
partially false	0.1654	0.3929	0.2328	56
true	0.8889	0.0381	0.0731	210
accuracy	0.5065	0.5065	0.5065	
macro avg	0.4128	0.3300	**0.2550**	612
weighted avg	0.6274	0.5065	0.4140	612

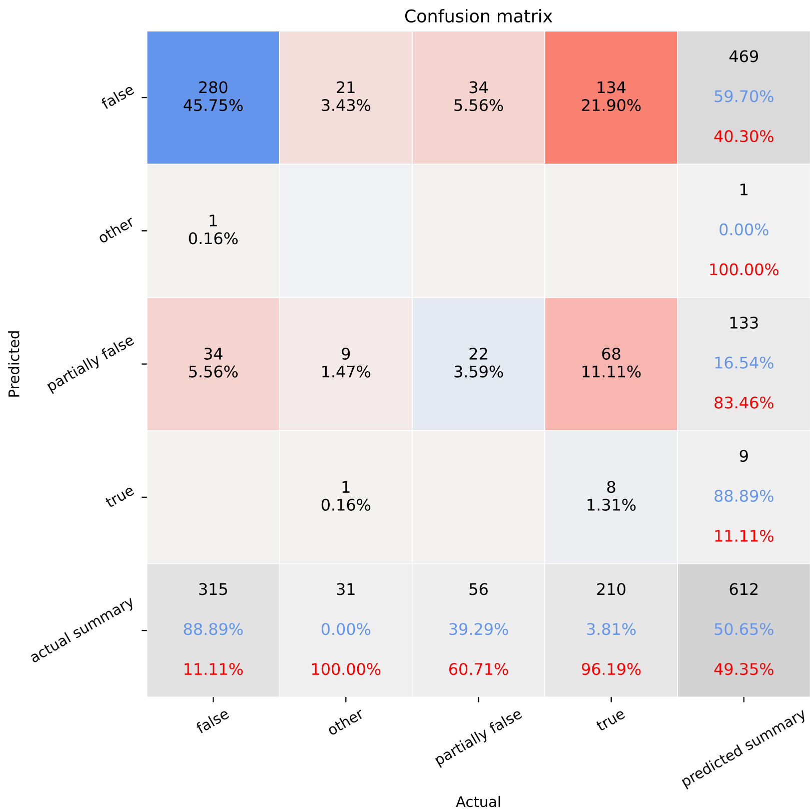


Figure 8: Confusion matrix for Task 3A with the original training dataset on the gold standard.

challenge. This is clearly reflected in Figure 7. The poor performance, especially for the categories *partially false* and *other*, correlates with the distribution of training data across these categories (see Figure 2b).

A commonly used tactic to deal with imbalanced datasets is to assign weights to each label. Alternative solutions for coping with unbalanced datasets for supervised machine learning are

Table 6
Classification report for the predictions on the original training data with oversampling against the gold standard.

	precision	recall	f1-score	support
false	0.5933	0.7873	0.6767	315
other	0.1667	0.0323	0.0541	31
partially false	0.1566	0.4643	0.2342	56
true	0.6818	0.0714	0.1293	210
accuracy	0.4739	0.4739	0.4739	
macro avg	0.3996	0.3388	**0.2736**	612
weighted avg	0.5621	0.4739	0.4168	612

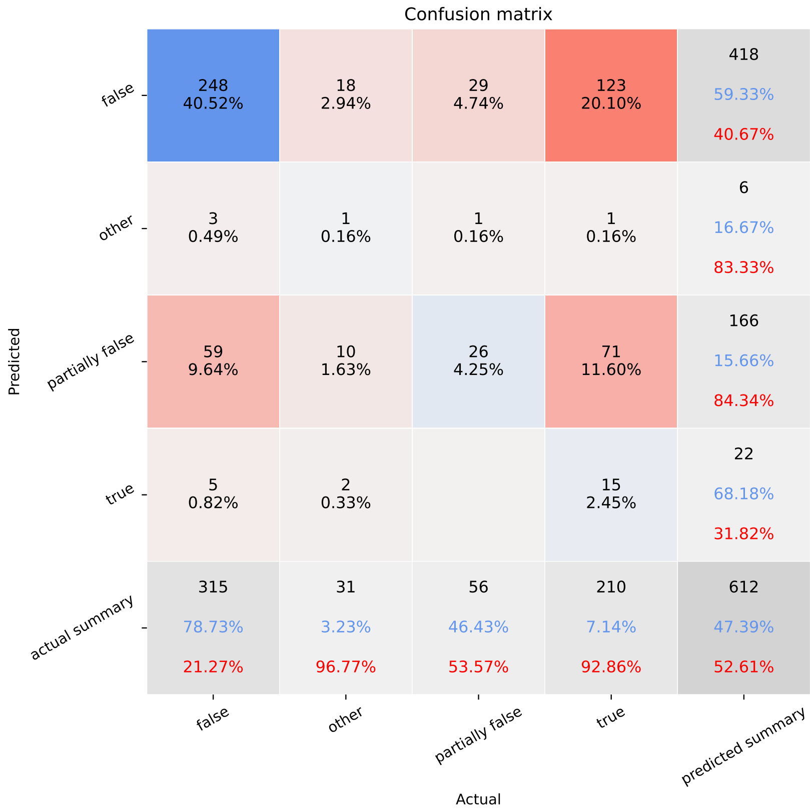


Figure 9: Confusion matrix for Task 3A with the original training dataset with oversampling on the gold standard.

undersampling or oversampling. Undersampling only considers a subset of an overpopulated class to end up with a balanced dataset. With the same goal, oversampling creates copies of the unbalanced classes. The influence of oversampling is evident from a comparison of both experiments on the original training data set (cf. Table 5 and 6). Thus, the macro-averaged F1 score was improved from 0.2550 to 0.2736.

Rank	Team	Accuracy	F1-macro
1	iCompass	0.5474	0.3391
2	nlpiruned	0.5408	0.3325
3	awakened	0.5310	0.3231
4	UNED	0.5441	0.3154
5	**NLytics**	**0.5131**	**0.3076**
6	SCUoL	0.5261	0.3047
7	hariharanrl	0.5359	0.2980
8	CIC	0.4755	0.2859
9	ur-iw-hnt	0.5327	0.2833
10	BUM	0.4722	0.2760
11	boby232	0.4755	0.2754
12	HBDCI	0.5082	0.2734
13	DIU_SpeedOut	0.5212	0.2707
14	DIU_Carbine	0.4722	0.2579
15	CODE	0.4444	0.2550
16	MNB	0.5065	0.2507
17	subMNB	0.5065	0.2507
18	fosil	0.4624	0.2505
19	Text_Minor	0.3775	0.2347
20	DLRG	0.5131	0.1987
21	DIU_Phoenix	0.2778	0.1593
22	AIT_FHSTP	0.1993	0.1549
23	DIU_SilentKillers	0.2598	0.1529
24	DIU_Fire71	0.2745	0.1328
25	AI Rational	0.0980	0.1165

Table 7
Results on Task 3A

Overfitting poses the most difficult challenge in these experiments and reduces generalizability. In all three experiments, we observe the same pattern of misclassification, which is due to difficulties of the system to find discriminative features (cf. Figure 7, 8, 9). The problem is most evident in the the poor performance of assigning the class label “true” on the test set. Most assigments were lost either to “false” and “partially false”. This issue is potentially caused by flaws in the selection of the training data. Indeed, we can attribute part of this problem to content features. At its most basic level, there is a significant difference in the average document length of the documents used for training and prediction, respectively. Following Table 2, significantly shorter documents were used for the training. The phenomenon is particularly evident for the category “true” (cf. Table 8). To support this hypothesis, however, the high standard deviation in both statistics suggests further investigation into outliers, as median and quantiles suggest a smaller deviation between test and training data.

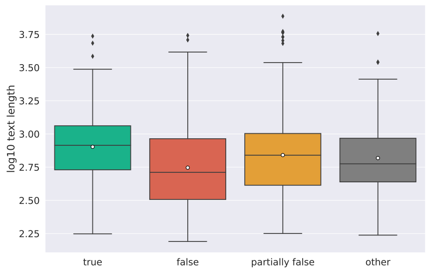
Further investigation examining lexical properties at the class level do not reflect significant differences in the training and testing data (cf. Figure 10). Even the use of the much larger data set does not effect the overall pattern (see Figure 7).

In fact, the problem may be due to a questionable choice of categories reflected in the class

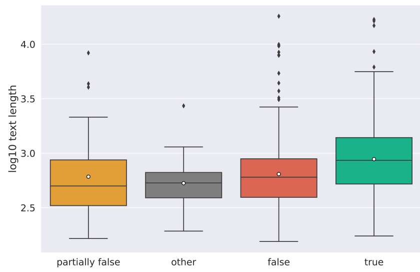
Table 8
Statistical summary of token (word) counts on the training set.

class		original training data	test data
false	doc count	493	315
	mean	760.81	1063.53
	std	739.57	1898.29
	min	17	68
	25%	341.00	400.00
	50%	512.00	671.00
	75%	969.00	1001.00
	max	6367	22168
partially false	doc count	313	56
	mean	984.60	1037.88
	std	1159.67	1542.03
	min	10	120
	25%	382.00	361.75
	50%	701.00	556.50
	75%	1094.00	954.00
	max	8751	10108
true	doc count	196	210
	mean	**1084.78**	**1481.96**
	std	894.23	2349.92
	min	123	60
	25%	493.25	533.00
	50%	890.00	968.00
	75%	1269.75	1552.00
	max	6064	19575
other	doc count	94	31
	mean	821.00	665.55
	std	902.36	512.82
	min	15	114
	25%	389.75	443.50
	50%	608.50	554.00
	75%	933.00	747.00
	max	6341	3005

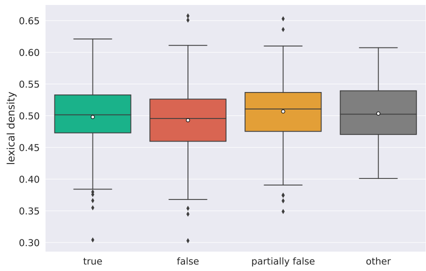
labels. In the case of the given task, the classification results suggest some kind of fact check. The system, however, is supposed to determine a truth value for an unseen document based solely on the available training data. We assume that in most cases external features contribute to the determination of the truth value of a certain statement. In particular, an individual's – this holds true for the sender as well as the receiver – worldview, contextual knowledge, and thematic context are crucial to their own decision. For this reason, linguistic means alone do not have enough discriminative power to robustly determine the truth value. Our approach is an attempt to narrow down the problem of distinguishing different views on a specific topic. Depending on the topic under investigation, we noticed significant differences in the performance of the trained systems with f1-scores ranging from 0.07 to 0.72.



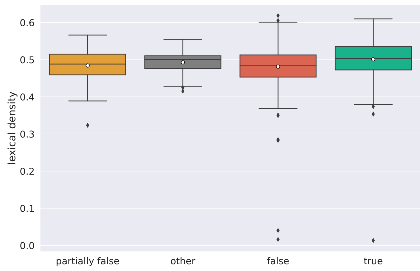
(a) log10 text length - training set



(b) log10 text length - test set



(c) lexical density - training set



(d) lexical density - test set

Figure 10: Class-based lexical feature comparison.

With the above findings, we achieve state of the art performance on the text classification datasets. Transformer-based models such as RoBERTa or Longformer have proven to be powerful language representation model for various natural language processing tasks. As this study shows, they are also an effective tool for multi-class text classification. In the future, we will further investigate the inner workings of Transformer-based models and how to counteract their tendency to overfitting.

6. Conclusion and Future work

In future work, we plan to investigate more recent neural architectures for language representation such as T5 [37], GPT-3 [38], or its open competitor OPT-175B [39].

Furthermore, we expect great opportunities for transfer learning from the areas such as argumentation mining [40] and offensive language detection [41]. In order to deal with data scarcity as a general challenge in natural language processing, we examine the application of concepts such as active learning, semi-supervised learning [42] as well as weak supervision [43]. With the evaluation of feature importance [44] we will further address the issue of robustness of our system, by explaining the individual features of the training data as well as their relevance to the models prediction.

References

[1] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, J. M. Struß, T. Mandl, R. Míguez, T. Caselli, M. Kutlu, W. Zaghouani, C. Li, S. Shaar, G. K. Shahi, H. Mubarak, A. Nikolov, N. Babulkov, Y. S. Kartal, J. Beltrán, The clef-2022 checkthat! lab on fighting the covid-19 infodemic and fake news detection, in: M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, V. Setty (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham, 2022, pp. 416–428.

[2] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, J. M. Struß, T. Mandl, R. Míguez, T. Caselli, M. Kutlu, W. Zaghouani, C. Li, S. Shaar, G. K. Shahi, H. Mubarak, A. Nikolov, N. Babulkov, Y. S. Kartal, J. Beltrán, M. Wiegand, M. Siegel, J. Köhler, Overview of the CLEF-2022 CheckThat! lab on fighting the COVID-19 infodemic and fake news detection, in: Proceedings of the 13th International Conference of the CLEF Association: Information Access Evaluation meets Multilinguality, Multimodality, and Visualization, CLEF '2022, Bologna, Italy, 2022.

[3] J. Köhler, G. K. Shahi, J. M. Struß, M. Wiegand, M. Siegel, T. Mandl, Overview of the CLEF-2022 CheckThat! lab task 3 on fake news detection, in: Working Notes of CLEF 2022—Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, volume 2017-Decem, 2017, pp. 5999–6009. arXiv:1706.03762.

[5] 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, 2019. arXiv:1907.11692.

[6] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The Long-Document Transformer (2020). arXiv:2004.05150.

[7] X. Zhou, R. Zafarani, Fake News: A Survey of Research, Detection Methods, and Opportunities, ACM Comput. Surv 1 (2018). arXiv:1812.00315.

[8] C. H. de Vreese, F. Esser, T. Aalberg, C. Reinemann, J. Stanyer, Populism as an Expression of Political Communication Content and Style: A New Perspective, International Journal of Press/Politics 23 (2018) 423–438. URL: http://journals.sagepub.com/doi/10.1177/1940161218790035. doi:10.1177/1940161218790035.

[9] U. Schade, F. Meißner, A. Pritzkau, S. Verschitz, Prebunking als Möglichkeit zur Resilienzsteigerung gegenüber Falschinformationen in Online-Medien, in: N. Zowislo-Grünewald, N. Wörmer (Eds.), Kommunikation, Resilienz und Sicherheit, Konrad-Adenauer-Stiftung, Berlin, 2021, pp. 134–155.

[10] J. Leskovec, K. Lang, Statistical properties of community structure in large social and information networks, Proceedings of the 17th international conference on World Wide Web. ACM (2008) 695–704. URL: http://dl.acm.org/citation.cfm?id=1367591.

[11] P. Badjatiya, S. Gupta, M. Gupta, V. Varma, Deep learning for hate speech detection in tweets, in: 26th International World Wide Web Conference 2017, WWW 2017 Companion, International World Wide Web Conferences Steering Committee, 2017, pp. 759–760. doi:10.1145/3041021.3054223. arXiv:1706.00188.

[12] L. Gao, R. Huang, Detecting online hate speech using context aware models, in: In-

ternational Conference Recent Advances in Natural Language Processing, RANLP, volume 2017-Septe, Association for Computational Linguistics (ACL), 2017, pp. 260–266. doi:10.26615/978-954-452-049-6-036. arXiv:1710.07395.

[13] J. Pavlopoulos, P. Malakasiotis, I. Androutsopoulos, Deeper attention to abusive user content moderation, in: EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings, Association for Computational Linguistics, Stroudsburg, PA, USA, 2017, pp. 1125–1135. URL: http://aclweb.org/anthology/D17-1117. doi:10.18653/v1/d17-1117.

[14] G. K. Pitsilis, H. Ramampiaro, H. Langseth, Effective hate-speech detection in Twitter data using recurrent neural networks, Applied Intelligence 48 (2018) 4730–4742. doi:10.1007/s10489-018-1242-y. arXiv:1801.04433.

[15] Z. Zhang, D. Robinson, J. Tepper, Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 10843 LNCS, Springer Verlag, 2018, pp. 745–760. doi:10.1007/978-3-319-93417-4_48.

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2018). arXiv:1810.04805.

[17] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding, Technical Report, 2019. arXiv:1906.08237.

[18] T. Mikolov, Q. V. Le, I. Sutskever, Exploiting Similarities among Languages for Machine Translation (2013). arXiv:1309.4168.

[19] J. Pennington, R. Socher, C. D. Manning, GloVe: Global vectors for word representation, in: EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014, pp. 1532–1543. doi:10.3115/v1/d14-1162.

[20] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017 - Proceedings of Conference, volume 2, 2017, pp. 427–431. URL: https://github.com/facebookresearch/fastText. doi:10.18653/v1/e17-2068. arXiv:1607.01759.

[21] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, Association for Computational Linguistics (ACL), 2018, pp. 2227–2237. doi:10.18653/v1/n18-1202. arXiv:1802.05365.

[22] M. Castelle, The Linguistic Ideologies of Deep Abusive Language Classification, 2019, pp. 160–170. doi:10.18653/v1/w18-5120.

[23] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl, The CLEF-2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News, in: Proceedings of the 43rd European Conference on Information Retrieval, ECIR˜'21, Lucca, Italy, 2021, pp. 639–649. URL: https://link.springer.com/chapter/10.1007/978-3-030-72240-1_75.

[24] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, N. Babulkov, A. Nikolov, G. K. Shahi, J. M. Struß, T. Mandl,

S. Modha, M. Kutlu, Y. S. Kartal, Overview of the CLEF-2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News, in: Proceedings of the 12th International Conference of the CLEF Association: Information Access Evaluation Meets Multiliguality, Multimodality, and Visualization, CLEF˜'2021, Bucharest, Romania (online), 2021.

[25] G. K. Shahi, J. M. Struß, T. Mandl, Overview of the CLEF-2021 CheckThat! Lab Task 3 on Fake News Detection, in: Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum, CLEF˜'2021, Bucharest, Romania (online), 2021.

[26] G. K. Shahi, J. M. Struß, T. Mandl, Task 3: Fake News Detection at CLEF-2021 CheckThat!, CLEF˜'2021, Zenodo, Bucharest, Romania (online), 2021. doi:10.5281/zenodo.4714517.

[27] G. K. Shahi, AMUSED: An Annotation Framework of Multi-modal Social Media Data (2020). arXiv:2010.00502.

[28] G. K. Shahi, A. Dirkson, T. A. Majchrzak, An exploratory study of covid-19 misinformation on twitter, Online Social Networks and Media 22 (2021) 100104.

[29] K. Shu, Fake News Detection Challenge KDD 2020, 2020. URL: https://www.kaggle.com/competitions/fakenewskdd2020/data.

[30] J. Ribeiro, Fakenews Classification Datasets, 2020. URL: https://www.kaggle.com/datasets/liberoliber/onion-notonion-datasets.

[31] O. Blanc, A. Pritzkau, U. Schade, M. Geierhos, CODE at CheckThat! 2022: Multi-class fake news detection of news articles with BERT, in: CEUR Workshop Proceedings, 2022. URL: http://ceur-ws.org.

[32] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research 3 (2003) 993–1022. URL: http://www.crossref.org/jmlr{_}DOI.html. doi:10.1162/jmlr.2003.3.4-5.993.

[33] M. Röder, A. Both, A. Hinneburg, Exploring the space of topic coherence measures, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, WSDM '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 399–408. URL: https://doi.org/10.1145/2684822.2685324. doi:10.1145/2684822.2685324.

[34] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-Art Natural Language Processing, in: arxiv.org, 2020, pp. 38–45. URL: https://github.com/huggingface/. doi:10.18653/v1/2020.emnlp-demos.6. arXiv:1910.03771v5.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, volume 32, Neural information processing systems foundation, 2019. URL: http://arxiv.org/abs/1912.01703. arXiv:1912.01703.

[36] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, J. Dean, Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation (2016). arXiv:1609.08144.

[37] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, arXiv 21 (2019) 1–67. arXiv:1910.10683.
[38] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.
[39] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin, et al., Opt: Open pre-trained transformer language models, arXiv preprint arXiv:2205.01068 (2022).
[40] M. Stede, Automatic argumentation mining and the role of stance and sentiment, Journal of Argumentation in Context 9 (2020) 19–41. URL: https://www.jbe-platform.com/content/journals/10.1075/jaic.00006.ste. doi:10.1075/jaic.00006.ste.
[41] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Predicting the type and target of offensive posts in social media, in: NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, volume 1, Association for Computational Linguistics, Stroudsburg, PA, USA, 2019, pp. 1415–1420. URL: http://aclweb.org/anthology/N19-1144. doi:10.18653/v1/n19-1144. arXiv:1902.09666.
[42] S. Ruder, B. Plank, Strong Baselines for Neural Semi-supervised Learning under Domain Shift, ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers) 1 (2018) 1044–1054. arXiv:1804.09530.
[43] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, C. Ré, Snorkel: rapid training data creation with weak supervision, in: VLDB Journal, volume 29, Springer, 2020, pp. 709–730. doi:10.1007/s00778-019-00552-1.
[44] S. M. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.