Making Sense of Nonsense: Integrated Gradient-based Input Reduction to Improve Recall for Check-worthy Claim Detection

Ghazaal Sheikhi[1,2], Andreas L. Opdahl[1], Samia Touileb[1] and Vinay Setty[2]

[1]*University of Bergen*
[2]*CHECK24 GmbH*
[2]*University of Stavanger*

Abstract

Analysing long text documents of political discourse to identify check-worthy claims (claim detection) is known to be an important task in automated fact-checking systems, as it saves the precious time of fact-checkers, allowing for more fact-checks. However, existing methods use black-box deep neural NLP models to detect check-worthy claims, which limits the understanding of the model and the mistakes they make. The aim of this study is therefore to leverage an explainable neural NLP method to improve the claim detection task. Specifically, we exploit well known integrated gradient-based input reduction on textCNN and BiLSTM to create two different *reduced claim data sets* from ClaimBuster. We observe that a higher recall in check-worthy claim detection is achieved on the data reduced by BiLSTM compared to the models trained on claims. This is an important remark since the cost of overlooking check-worthy claims is high in claim detection for fact-checking. This is also the case when a pre-trained BERT sequence classification model is fine-tuned on the reduced data set. We argue that removing superfluous tokens using explainable NLP could unlock the true potential of neural language models for claim detection, even though the reduced claims might make no sense to humans. Our findings provide insights on task formulation, design of annotation schema and data set preparation for check-worthy claim detection.

Keywords

claim detection, check-worthy claims, fact checking, input reduction

1. Introduction

With the rise of concerns around misinformation threatening democracy and freedom in recent decades, fact-checking has become an integral part of journalism. Fact-checking is an extensively burdensome procedure as it requires a sequence of rigorous tasks including identifying check-worthy claims, monitoring related fact-checks, collecting reliable evidence, verifying the asserted facts, and publishing the fact-check [1]. Considering the volume and the speed of dissemination of misleading content in today's digital era, it is a demanding task for the fact-checking community with its limited resources and manpower. Automated fact-checking (AFC) technologies could evidently assist in expediting and scaling-up the process.

NAIS 2023: The 2023 symposium of the Norwegian AI Society, June 14-15, 2023, Bergen, Norway.
ghazaal.sheikhi@gmail.com (G. Sheikhi); andreas.opdahl@uib.no (A. L. Opdahl); samia.touileb@uib.no (S. Touileb); vinay.j.setty@uis.no (V. Setty)

In recent years, the breakthrough in Natural Language Processing (NLP) due to pre-trained neural language models has led to a rapid growth in AFC-related research [2, 1, 3]. Several end-to-end fact-checking systems have been proposed in the literature with promising results in the experimental setting [4, 5, 6], and yet perform drastically poorly in practice [7, 8]. According to a survey on AFC landscape by Reuters Institute for the Study of Journalism, fully automated verification is an unattainable goal with today's technology as fact-checkers rely on their knowledge about the context, expertise, and unbiased judgment to verify a claim [7]. On the other hand, studies on the user needs of fact-checkers show that monitoring social media and political sources (to identify and rank claims to check) receive the highest preference among other AFC tools [7, 8].

Automated claim detection is formalized as either a ranking or a classification problem, where models are trained on data sets of long text parsed into sentences and labelled or ranked by humans according to their check-worthiness [9]. There is solid evidence from various perspectives that check-worthiness is rather associated with features manifested in specific *spans* in the claim such as numerical values, past tense verbs, causation and prediction [9, 10, 11, 12]. However, to the best of our knowledge, there is no claim detection data set with span labels.

On the other hand, gradient-based explainable NLP has been studied [13, 14, 15] for attributing the predictions to input features [16, 17]. This has motivated us to exploit the neural models to reduce the inputs into minimal *spans/tokens* and inspect the behaviour of the claim detection systems when trained on the reduced claims. To the best of our knowledge, this is the first study of integrated gradient-based input reduction for claim detection. The aim here is not to expose the nonsense in predictions, but to make sense of this nonsense. We contribute to the AFC landscape by showcasing how explainable NLP methods could improve automated claim detection. Our findings suggest that the problem of check-worthiness detection could be extended to pivotal span/token identification. Another reflection of this work is providing insights on the future of AFC on how to build claim detection data sets, interlaced with our understanding of the model behaviour.

The rest of this paper is organized as follows. In Section 2, we review related studies on automated claim detection. We explain and give details about our methodology in Section 3. Section 4 presents the experiments conducted on ClaimBuster data set [18]. Finally, we conclude the paper in Section 5 by summarizing our main finding and discussing possible future works.

2. Related Work

One of the first studies on claim detection for automated fact-checking was the work by Hassan et al. [19] who employed traditional classification models to recognize check-worthy factual statements (CFS) from non-factual statements (NFS) and unimportant factual statements (UFS) in a data set of U.S. presidential debates [19]. This work was extended later to ClaimBuster, a fact-checking platform with a claim spotting component based on NLP and supervised learning [9]. Their finding suggest that the two most discriminating features among sentiment, word count, TF-IDF part of speech (PoS) tags and named entity (NE) types are two PoS tags: those that mark the past tense of a verb and those that mark cardinal numbers. A few years later in 2020, ClaimBuster data set of human-labeled claims extracted from U.S. presidential debates (1960-

2016) was published [18].

The check-worthiness detection sub-tasks in CLEF CheckThat! editions (introduced in 2018 and ongoing) grew more interest among the NLP researchers on claim detection [20, 21]. Different editions of CheckThat! Lab offer data sets in different languages including English, Turkish, Arabic, Bulgarian, and Spanish for detecting check-worthy claims on Twitter and political debates. Multiple teams have participated in the challenges by proposing solutions mostly based on neural language models. For instance, the top ranked teams in CheckThat! 2020 adopted BERT [22] and RoBERTa [23] with enhanced generalization capability [24] to spot check-worthy Tweets. For the task of detecting claims in political debates, none of the solutions could beat the naive BiLSTM [25] model with GloVe embedding [26]. The problem of claim detection from Twitter was also addressed by BERT and RoBERTa models enhanced by augmenting training data with synthetic check-worthy claims generated by lexical substitutions using BERT-based embedding [20]. Data augmentation by substitutions using WordNet was also employed by the top-ranked team in claim detection from political debates [20]. Their BERTweet model fine-tuned on normalized and augmented claims surpassed the reference n-gram model.

More recently, the prominence of the factors and features central to check-worthiness has started to be noted in the literature. Kartal et al. [27] have incorporated domain specific controversial topics compiled from Wikipedia as well as presence of comparative/superlative adjectives to their logistic regression model [27]. FactRank, the first claim detection tool for the Dutch language, operates based on a convolutional neural network (CNN) sentence classifier [28]. They have firstly developed a detailed code-book created by expert fact-checkers through an iterative process to define the "concept of check-worthiness" and to guide the annotators in identifying check-worthy claims. Konstantinovskiy et al. [12] also points out the complexity of claim detection and the inconsistency among human annotators [12]. They have developed an annotation schema to define seven categories of claims and generated a dataset of $5,571$ labelled sentences. The consistency of the schema has been evaluated by a claim detection system based on sentence representations concatenated with PoS and NE tags, and a logistic regression classifier. Alhindi et al. [29] have presented a data set of news articles on climate change and shown that augmenting argumentative discourse structure of claims annotated in the data improves the performance of the claim detection [29].

Reformulating the claim detection problem by incorporating more attributes such as the claimer, their stance, the topic, the claim object, and the claim spans is the idea behind the NewsClaims [30]. The NewsClaims data set annotated for the corresponding attributes was released as part of their work. Claim span detection identifies the boundaries of what is called *the actual claim*. These studies, despite seeming as disparate solutions for the claim detection problem, allude to the idea that perhaps a subset of tokens in a claim bear check-worthiness information whether they are particular NEs or a contiguous sequence of tokens. However, there is a gap in the literature on how NLP models behave when the input is reduced to a subset of tokens not necessarily sensible to humans.

On the other hand, as the neural language models become widespread, interpretability methods serves to provide post-hoc explanations about model predictions. There are several categories of interpretability approaches such as LIME, HotFLIP, adversarial examples, etc. [13]. An important category of these approaches is input feature explanation [13]. These methods are adaptable to different models and provide a human understandable explanation on how

important the words/tokens are for a specific input [13]. A straightforward technique to input feature explanation is based on gradient [16, 13]. Gradient-based explainable NLP has been studying in rationalizing the neural predictions [31] and uncovering erroneous logic in the models [17]. However, we are not aware of any study that has focused on gradient-based input reduction to improve classification.

3. Methodology

In this work, we assess the performance of three NLP models, namely textCNN [32], BiLSTM [25], and BERT [33] when trained on the claims reduced to a subset of tokens. Input reduction is one of the adversarial attacks used to investigate on the pathologies of neural language models and reveal the pivotal tokens behind some of their predictions [14]. It works by iterative removal of the least important tokens from the input sequence until the model prediction changes. The reduced input is thus presumed to be an important part of the input sequence that is critical for the prediction.

Input reduction requires a technique to measure the importance of the tokens in the input sequence. To identify the unimportant tokens, we use the attribution method layer integrated gradients [16]. Feature attribution refers to a set of non-adversarial techniques for interpreting deep neural networks. The attribution score measures the contribution of each of the input dimensions to the model prediction. Integrated gradient is one of the most efficient attribution methods with straightforward computation and no requirement for instrumentation of the model [16]. In deep neural models, layer integrated gradient computes an attribution score for each dimension of the token embedding vector.

Assume that the neural network model is denoted by a function $F : R^n \rightarrow [0, 1]$ with $x \in R^n$, $x' \in R^n$ and $F(x)$ being the input, the reference input and the output of the network respectively. The input is the embedding vector and the reference input is simply a zero embedding vector. The gradient is calculated as the line integral of the gradients through the straight line path in R^n connecting input to the reference input. For the i^{th} dimension of the given input, x, the integrated gradient is defined as $IG_i(x)$ as follows [16, 17].

$$IG_i(x) = (x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha,$$

where, $\frac{\partial F(x)}{\partial x_i}$ refers to the gradient of $F(x)$ along the i^{th} dimension of x.

The integral in the formula could be approximated by the summation operator across the points on the line connecting x to x with sufficiently small distance [16].

$$\tilde{IG}_i(x) = \frac{(x_i - x'_i)}{m} \times \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i},$$

where m denotes the number of steps in the Riemann approximation of the integral.

Table 1
Hyperparameter setting for textCNN and BiLSTM for claim reduction

textCNN	**BiLSTM**
max_len = 50	max_len = 50
embedding_size = 300	embedding_size = 300
learning_rate = 0.001	learning_rate = 0.001
n_epochs = 15	n_epochs = 10
batch_size = 512	batch_size = 256
dropout = 0.5	dropout = 0.1

For a given claim, the token attribution score is obtained by summing up the attribution scores across all embedding dimensions. The network is trained on the training split and the reduced data is obtained from the unseen validation set. During the validation phase, the tokens are iteratively removed from the input claim until the prediction is flipped. The minimal sequence of tokens that maintains the predicted label is stored. This way, we ensure that the true labels are not exposed during the reduction.

To form the reduced data set, we employ textCNN and BiLSTM. For the purpose of our study, i.e. claim reduction, we prefer these simple sequence classification models over the more powerful pre-trained models because the former embed less linguistic and other extraneous information that could impact our results. Therefore, the reduced data set is formed purely based on the patterns learnt from the input data. We use $50\% - 50\%$ as a train-validation split, where the model is trained on the training split and the reduced claims are attained from the validation split. Since we require deriving the reduced claims for the whole data, the train-test splits are then swapped to get the reduced claims for the formerly training split. The results are then concatenated to form the reduced data set. Admittedly, merging the results from two models trained on different subsets of the data could be questioned. But it is an acceptable framework to roughly generate the reduced claim data sets for the purpose of this study.

4. Experiments and Results

We conducted a set of experiments on the ClaimBuster claim detection dataset, which consists of the statements from the transcripts of the U.S. presidential debates [18]. The sequences are labelled as CFS, NFS and UFS. We use a set of the data which according to the publisher [18] has been labelled under stricter criteria and is supposedly of higher quality for building the models. This set includes in total $11,056$ statements, where $8,292$ are NFS and $2,764$ are CFS.

We use textCNN and BiLSTM for claim reduction and then detection. Pre-trained BERT is also fine-tuned for the claim detection tasks.

4.1. Claim reduction

To implement the layer integrated gradient, the open source Captum library for model interpretation is deployed. To generate the reduced claim data sets, PyTorch nn.Module is used to

Table 2
Examples from reduced data sets generated by our two models, textCNN and BiLSTM.

Claim	Label	Reduced claim	
		textCNN	BiLSTM
'saddam hussein was a risk to our country, madam.'	NFS	'saddam hussein'	''
'and we won the nomination by going out into the streets - barbershops, beauty parlors, restaurants, stores, in factory shift lines also in farmers' markets and livestock sale barns - and we talked a lot, and we listened a lot and we learned from the american people.'	NFS	'into the streets stores shift lines also in markets and'	'the'
'and i favor a shifting of the welfare cost away from the local governments altogether.'	NFS	''	''
'secondly, in haiti, political violence is much, much smaller than it was.'	CFS	''	'secondly in haiti political violence much smaller than was'
'he did not take that position on tibet.'	CFS	''	''
'we have spent over $600 billion so far, soon to be $1 trillion'	CFS	'600'	'we have spent over 600 to be trillion'
'you know, because it sounds like you are in the business, or you are aware of people in the business – you know that we are now for the first time ever energy-independent.'	CFS	'you'	'sounds'

Table 3
The number of NFS and CFS statements reduced to empty sequences by textCNN and BiLSTM.

Label	Total	Reduced to empty sequence	
		textCNN	BiLSTM
NFS	8,292	6,338	5,812
CFS	2,764	484	508

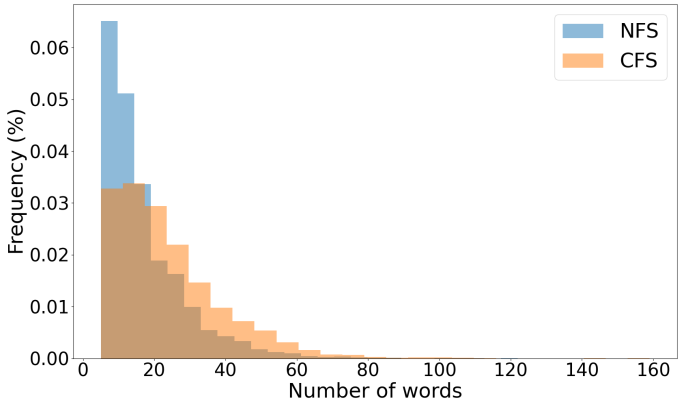
implement textCNN and BiLSTM. The data set is split into $50\% - 50\%$ Train-test set. For both models, cross entropy loss is used, and the weights are optimized with Adam [34].

In the textCNN architecture, we use filter_sizes = $[4, 5, 6]$ and num_filters = 64. In the BiLSTM architecture, we set the hidden_size = 256 with average and max pooling. There is a linear layer of size $(hidden\_size \times 4, 64)$, a ReLU layer, and a linear output layer of size $(64, 2)$.

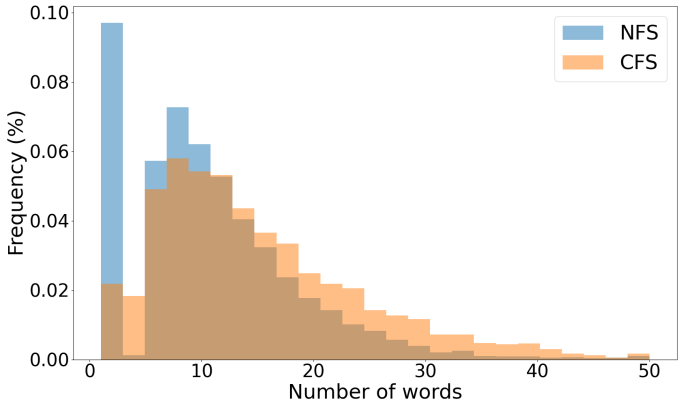
The other hyperparameter configurations of the models are shown in Table 1.

Two reduced data sets are formed by textCNN and BiLSTM. Table 2 illustrates a few examples of these reduced data sets. We observe that the majority of the NFS statements are reduced to empty sequences by both models (See Table 3). This is in line with our earlier argument on CFSs

Figure 1: The probability densities of the number of words in NFS and CFS sequences.



(a) Claims



(b) Reduced claims by BiLSTM

being characterized by particular token/spans and not NFSs. An empty sequence is indeed an NFS, but this phenomenon hinders further experiments. To tackle this issue, two replacement scenarios were tested: substituting the empty entries with the unknown token *[UNK]* or with the claim. As the *[UNK]* scenario resulted in extremely poor quality models, we consented to the latter solution. It is important to ensure that we are not reducing the problem of distinguishing between NFS and CFS to identifying long grammatical statements from short sequences of tokens. Figure 1 illustrates the probability densities of the number of words in NFS and CFS sequences for both claims and the reduced claims by BiLSTM. These plots visually confirm that there is not a remarkable difference between the distributions in the two settings that would drastically affect the system. A similar pattern is observed for claims reduced by textCNN.

Table 4
Hyperparameter setting for textCNN, BiLSTM and BERT for claim detection

textCNN/BiLSTM	**BERT**
max_len = 50	-
embedding_size = 300	-
learning_rate = 0.01	learning_rate = 2e-5
optimizer = Adam	optimizer = AdamW
n_epochs = 15	n_epochs = 2
batch_size = 512	batch_size = 8
dropout = 0.5	hidden_dropout = 0.5
-	attention_dropout= 0.5
-	classifier_dropout = 0.2
-	weight_decay = 0.01

Table 5
Claim detection results by textCNN and BiLSTM on claims and reduced claims.

Model	**Data**	**F1**	**Precision**	**Recall**
textCNN	Claims	0.719 (0.013)	**0.727** (0.021)	0.743 (0.023)
	Reduced by textCNN	0.659 (0.013)	0.712 (0.017)	0.630 (0.029)
	Reduced by BiLSTM	**0.723** (0.003)	0.662 (0.015)	**0.797** (0.017)
BiLSTM	Claims	0.766 (0.007)	**0.779** (0.017)	0.762 (0.014)
	Reduced by textCNN	0.701 (0.003)	0.728 (0.0206)	0.684 (0.021)
	Reduced by BiLSTM	**0.766** (0.010)	0.758 (0.015)	**0.775** (0.019)

4.2. Claim detection

Claim detection is a key component in AFC to reduce the search space for human fact-checkers. In this problem, the cost of overlooking the positive class i.e. CFS is higher than the cost of mistaking non-factual statement for CFS. As we rely on F1, precision, and recall in our experiments to evaluate the performance of the models, recall should be more emphasized.

The first claim detection experiments are based on textCNN and BiLSTM models trained on the claim data set and the reduced data sets. We conduct 5 runs of 5-fold cross validation (80% train-20%-validation) with different random seed values. PyTorch nn.Module is used to implement textCNN and BiLSTM for claim detection. For these two models, we conduct 5 runs of 5-fold cross validation ($80\% - 20\%$) with different random seed values. For both textCNN and BiLSTM models, cross entropy loss is used, and the weights are optimized with Adam [34].

In the textCNN architecture, we use filter_sizes = $[4, 5, 6]$ and num_filters = 128. In the BiLSTM architecture, we set the hidden_size = 256 with average and max pooling. There is a linear layer of size ($hidden\_size \times 4, 64$), a ReLU layer, and a linear output layer of size ($64, 2$).

The BERT claim detection model is implemented using a pre-trained language model implemented by the HuggingFace `transformers` library [35]. The model is fine-tuned with AdamW [36] following a warmed-up. We split the data into train, validation, and test sets ($40\% - 40\% - 20\%$) and use the validation set to return the best model after two epochs

Table 6
Claim detection results by BERT on claims and reduced claims.

Data	Validation			Test		
	F1	Precision	Recall	F1	Precision	Recall
Claims	**0.826** (0.004)	**0.851** (0.024)	0.803 (0.026)	**0.820** (0.008)	**0.839** (0.022)	0.804 (0.027)
Reduced by BiLSTM	0.805 (0.014)	0.797 (0.031)	**0.815** (0.027)	0.800 (0.022)	0.789 (0.030)	**0.812** (0.023)

(We observed that usually one or two epochs are enough to attain the best-fitting model).

The other hyperparameter configurations of the models are shown in Table 4.

To make a fair comparison, the data splits, the number of epochs, and the model parameters maintain the same across all the runs. The validation scores in terms of F1, precision, and recall are averaged across all runs. The positive class is CFS. Table 5 presents the average scores and the standard deviations (numbers in parentheses). We observe that the data set of claims reduced by BiLSTM results in the highest F1 and recall for both models. The improvement in F1 is not remarkable though as the reduction significantly decreases the precision. The highest recall is achieved when textCNN is trained on the BiLSTM reduced data set. However, this scenario leads to a profoundly low precision. In terms of precision, the BiLSTM model trained on the claim data set leads to the highest score. The superiority of the BiLSTM reduced data set over textCNN is not surprising as this model is more competent at claim detection prior to reduction according to Table 5.

To further explore the potential of reduced data, the pre-trained distilled BERT model for sequence classification is fine-tuned on the claim data set and the BiLSTM reduced data set. We split the data into train, validation, and test sets (40%-40%-20%) and use the validation set to return the best model after two epochs (We observed that usually one or two epochs are enough to attain the best-fitting model). The tests are run for five times with different random seed values for the data split. Performance scores on validation and test splits shown in Table 6 are averaged over the five runs with standard deviations in parentheses. It is the case that the model trained on reduced data set exhibits significant drops in F1 and particularly in precision when compared to the model trained on the claims. However, the reduced data set results in a higher recall. We argue that reducing the input pushes the model predictions towards the positive class. Since we replaced the empty NFSs by the claims, there are long sequences of the negative class in the data and one might expect the model to learn the NFSs better. When this is not the case, then the reduced CFSs clearly contain some tokens central to the classification task.

5. Conclusion

In this study, we contribute to the AFC literature by inspecting the problem of claim detection from a novel perspective. We study how the faulty behaviour of NLP models i.e. generating predictions out of a subset of tokens nonsense to humans could be leveraged. Our experiments

confirm that the models trained on the artificially created reduced claim data sets result in a higher recall compare to the models trained on the original claim data set. We argue that extending the claim detection task to the task of pivotal span/token identification while considering the behaviour of NLP models could lead to a better performance. We believe that this work provides insights into reformulating the task of claim detection, designing annotation schemes, and preparing the data sets.

Limitations

The main limitations of this study are in the approach we followed to generate a set of reduced claims from the whole data set. We had to merge the two data sets obtained from the two models trained on different splits of the claim data set, that could result in inconsistencies. The second limitation is replacing the empty claims with the original claim. Although it is shown that the distribution of the length of claims are compatible, some heuristics could be devised to guide the reduction as a future direction. Claim reduction could be studied by taking into account the linguistic features of the claims. We will also consider analysing the reduced claims for sensible patterns and comparing the patterns with the annotation schemes studied in the literature.

Acknowledgments

We would like to thank TV2, Bergens Tidende, and Faktisk for their insightful remarks on the user needs in automated fact-checking.

This research was supported by industry partners and the Research Council of Norway with funding to MediaFutures: Research Centre for Responsible Media Technology and Innovation, through the Centres for Research-based Innovation scheme, project number 309339.

References

[1] X. Zeng, A. S. Abumansour, A. Zubiaga, Automated fact-checking: A survey, Language and Linguistics Compass 15 (2021) e12438. URL: https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/lnc3.12438. doi:https://doi.org/10.1111/lnc3.12438.

[2] Z. Guo, M. Schlichtkrull, A. Vlachos, A survey on automated fact-checking, Transactions of the Association for Computational Linguistics 10 (2022) 178–206. URL: https://aclanthology.org/2022.tacl-1.11. doi:10.1162/tacl_a_00454.

[3] E. Lazarski, M. Al-Khassaweneh, C. Howard, Using nlp for fact checking: A survey, Designs 5 (2021). doi:10.3390/designs5030042.

[4] N. Hassan, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, A. K. Nayak, V. Sable, C. Li, M. Tremayne, Claimbuster: The first-ever end-to-end fact-checking system, Proc. VLDB Endow. 10 (2017) 1945–1948. URL: https://doi.org/10.14778/3137765.3137815. doi:10.14778/3137765.3137815.

[5] S. Li, S. Zhao, B. Cheng, H. Yang, An end-to-end multi-task learning model for fact checking, in: Proceedings of the First Workshop on Fact Extraction and VERification

(FEVER), Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 138–144. URL: https://aclanthology.org/W18-5523. doi:10.18653/v1/W18-5523.

[6] S. Ahmed, K. Hinkelmann, F. Corradini, Fact checking: An automatic end to end fact checking system, in: M. Lahby, A.-S. K. Pathan, Y. Maleh, W. M. S. Yafooz (Eds.), Combating Fake News with Computational Intelligence Techniques, Springer International Publishing, Cham, 2022, pp. 345–366. URL: https://doi.org/10.1007/978-3-030-90087-8_17. doi:10.1007/978-3-030-90087-8_17.

[7] L. Graves, Understanding the promise and limits of automated fact-checking, Reuters Institute for the Study of Journalism (2018).

[8] L. Dierickx, G. Sheikhi, D. T. D. Nguyen, C.-G. Lindén, Report on the user needs of fact-chekers, in: NORDIS Project Report: University of Bergen, Task 4.2, 2022. URL: https://datalab.au.dk/fileadmin/Datalab/NORDIS_reports/Report_task_4.2_Fact-checkers_user_needs.pdf.

[9] N. Hassan, F. Arslan, C. Li, M. Tremayne, Toward automated fact-checking: Detecting check-worthy factual claims by claimbuster, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 1803–1812. URL: https://doi.org/10.1145/3097983.3098131. doi:10.1145/3097983.3098131.

[10] F. Arslan, J. Caraballo, D. Jimenez, C. Li, Modeling factual claims with semantic frames, in: Proceedings of the 12th Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 2511–2520. URL: https://aclanthology.org/2020.lrec-1.306.

[11] C. Zuo, A. I. Karakas, R. Banerjee, To check or not to check: Syntax, semantics, and context in the language of check-worthy claims, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 10th International Conference of the CLEF Association, CLEF 2019, Lugano, Switzerland, September 9–12, 2019, Proceedings, Springer-Verlag, Berlin, Heidelberg, 2019, p. 271–283. URL: https://doi.org/10.1007/978-3-030-28577-7_23. doi:10.1007/978-3-030-28577-7_23.

[12] L. Konstantinovskiy, O. Price, M. Babakar, A. Zubiaga, Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection, Digital Threats 2 (2021). URL: https://doi.org/10.1145/3412869. doi:10.1145/3412869.

[13] A. Madsen, S. Reddy, S. Chandar, Post-hoc interpretability for neural nlp: A survey, ACM Comput. Surv. (2022). URL: https://doi.org/10.1145/3546577. doi:10.1145/3546577, just Accepted.

[14] S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, J. Boyd-Graber, Pathologies of neural models make interpretations difficult, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 3719–3728. URL: https://aclanthology.org/D18-1407. doi:10.18653/v1/D18-1407.

[15] J. Wang, J. Tuyls, E. Wallace, S. Singh, Gradient-based analysis of NLP models is manipulable, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 247–258. URL: https://aclanthology.org/2020.findings-emnlp.24. doi:10.18653/v1/2020.findings-emnlp.24.

[16] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings

of the 34th International Conference on Machine Learning - Volume 70, ICML'17, JMLR.org, 2017, p. 3319–3328.

[17] P. K. Mudrakarta, A. Taly, M. Sundararajan, K. Dhamdhere, Did the model understand the question?, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 1896–1906. URL: https://aclanthology.org/P18-1176. doi:10.18653/v1/P18-1176.

[18] F. Arslan, N. Hassan, C. Li, M. Tremayne, A Benchmark Dataset of Check-worthy Factual Claims, in: 14th International AAAI Conference on Web and Social Media, AAAI, 2020.

[19] N. Hassan, C. Li, M. Tremayne, Detecting check-worthy factual claims in presidential debates, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 1835–1838. URL: https://doi.org/10.1145/2806416.2806652. doi:10.1145/2806416.2806652.

[20] S. Shaar, M. Hasanain, B. Hamdan, Z. S. Ali, F. Haouari, M. K. Alex Nikolov, F. A. Yavuz Selim Kartal, G. Da San Martino, A. Barrón-Cedeño, R. Míguez, T. Elsayed, P. Nakov, Overview of the CLEF-2021 CheckThat! lab task 1 on check-worthiness estimation in tweets and political debates, in: Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum, CLEF '2021, Bucharest, Romania (online), 2021. URL: http://ceur-ws.org/Vol-2936/paper-28.pdf.

[21] 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.

[22] G. S. Cheema, S. Hakimov, R. Ewerth, Check_square at checkthat! 2020 claim detection in social media via fusion of transformer and syntactic features, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_216.pdf.

[23] L. Zhuang, L. Wayne, S. Ya, Z. Jun, A robustly optimized BERT pre-training approach with post-training, in: Proceedings of the 20th Chinese National Conference on Computational Linguistics, Chinese Information Processing Society of China, Huhhot, China, 2021, pp. 1218–1227. URL: https://aclanthology.org/2021.ccl-1.108.

[24] E. M. Williams, P. Rodrigues, V. Novak, Accenture at checkthat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_226.pdf.

[25] M. Schuster, K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on

Signal Processing 45 (1997) 2673–2681. doi:10.1109/78.650093.
[26] J. R. Martinez-Rico, L. Araujo, J. Martínez-Romo, Nlp&ir@uned at checkthat! 2020: A preliminary approach for check-worthiness and claim retrieval tasks using neural networks and graphs, in: Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, 2020.
[27] Y. S. Kartal, M. Kutlu, B. Güvenen, Too many claims to fact-check: Prioritizing political claims based on check-worthiness, in: CEUR Workshop Proceedings, CEUR-WS, 2020.
[28] B. Berendt, P. Burger, R. Hautekiet, J. Jagers, A. Pleijter, P. V. Aelst, Factrank: Developing automated claim detection for dutch-language fact-checkers, Online Social Networks and Media 22 (2021). doi:10.1016/j.osnem.2020.100113.
[29] T. Alhindi, B. McManus, S. Muresan, What to fact-check: Guiding check-worthy information detection in news articles through argumentative discourse structure, in: Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, Association for Computational Linguistics, Singapore and Online, 2021, pp. 380–391. URL: https://aclanthology.org/2021.sigdial-1.40.
[30] R. G. Reddy, S. Chinthakindi, Z. Wang, Y. R. Fung, K. S. Conger, A. S. Elsayed, M. Palmer, H. Ji, Newsclaims: A new benchmark for claim detection from news with background knowledge, CoRR abs/2112.08544 (2021). URL: https://arxiv.org/abs/2112.08544. arXiv:2112.08544.
[31] T. Lei, R. Barzilay, T. Jaakkola, Rationalizing neural predictions, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 107–117. URL: https://aclanthology.org/D16-1011. doi:10.18653/v1/D16-1011.
[32] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 1746–1751. URL: https://aclanthology.org/D14-1181. doi:10.3115/v1/D14-1181.
[33] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.
[34] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014. URL: https://arxiv.org/abs/1412.6980. doi:10.48550/ARXIV.1412.6980.
[35] Huggingface, Distil BERT for sequence classification, https://huggingface.co/docs/transformers/model_doc/distilbert#transformers.DistilBertForSequenceClassification, 2022.
[36] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2017. URL: https://arxiv.org/abs/1711.05101. doi:10.48550/ARXIV.1711.05101.