Journal of Computational Social Science
https://doi.org/10.1007/s42001-024-00289-0

RESEARCH ARTICLE

Propositional claim detection: a task and dataset for the classification of claims to truth

Sami Nenno[1,2]

Received: 4 March 2024 / Accepted: 21 April 2024

Abstract

This paper introduces Propositional Claim Detection (PCD), an NLP task for classifying claims to truth, and presents a publicly available dataset for it. PCD is applicable in practical scenarios, for instance, for the support of fact-checkers, as well as in many areas of communication research. By leveraging insights from philosophy and linguistics, PCD is a more systematic and transparent version of claim detection than previous approaches. This paper presents the theoretical background for PCD and discusses its advantages over alternative approaches to claim detection. Extensive experiments on models trained on the dataset are conducted and result in an F_1-score of up to 0.91. Moreover, PCD's generalization across domains is tested. Models trained on the dataset show stable performance for text from previously unseen domains such as different topical domains or writing styles. PCD is a basic task that finds application in various fields and can be integrated with many other computational tools.

Keywords Misinformation · NLP · Claim detection · Truth-values

Introduction

Extracting meaning from text is a central part of communication research and in the recent years this is increasingly done with the aid of computational methods [1]. This paper introduces a (German-language) dataset for Propositional Claim Detection (PCD), a Natural Language Processing (NLP) task that aims at classifying sentences that can be true or false. Models trained on this dataset can be used for various purposes in communication research and beyond. One natural domain

✉ Sami Nenno
sami.nenno@hiig.de

[1] Centre for Media, Communication and Information Research, University of Bremen, Linzer Str. 4, 28359 Bremen, Bremen, Germany

[2] AI & Society Lab, Humboldt Institute for Internet and Society, Französische Str. 9, 10117 Berlin, Berlin, Germany

Published online: 09 May 2024

of application is misinformation. PCD is in line with previous approaches to claim detection that aim at the support of fact-checkers by identifying claims that potentially carry misinformation [2].

Sentences that can be true or false are known in philosophy and mathematical logic as sentences with propositional content. As propositional sentences are almost exclusively declarative sentences, PCD mostly leverages grammatical information (e.g., word order, punctuation, or tense) for classification. Grammatical or syntactical information has often been neglected in computational methods for social scientific research [3]. However, it has useful properties that are beneficial to methods like PCD. A core assumption of this paper is that the grammatical information that is leveraged by PCD remains relatively stable across domains and accordingly PCD-models can be used in various contexts and purposes. The guiding questions of this paper are if PCD can reach competitive results to other approaches to claim detection and if these results remain stable across texts of different topical domain, time periods or writing styles. As it will be shown, both questions have a positive answer.

This paper makes multiple contributions: (1) a new task design for the detection of claims is introduced, (2) a dataset for PCD is described and made publicly available, (3) an array of models is tested on the dataset with a special focus on domain adaptation.[1]

Background

Automated fact-checking

Guo et al. [4] provide a survey of different sub-tasks for automated fact-checking and find research on claim detection, retrieving previously checked claims, claim verification, and generating a textual justification for a verdict. Especially claim verification has received much attention (see e.g., [5, 6] for a survey on the related task of stance detection). For this task, a model is supposed to derive an evidence-based verdict (true, false, etc.) for a given claim. However, there has been critique on this task. Glockner et al. [7] argue that many approaches are developed and tested in an unrealistic setup and cannot refute real-world misinformation. Also, Nakov et al. [8] find that human fact-checkers have little trust in a fully-automated pipeline and rather find interest in tools like claim detection that only aim at partial automation.

Models for claim detection are trained to identify claims that potentially carry misinformation. The aim is to decrease the workload of human fact-checkers by providing a pre-selection of relevant claims that can then be verified. A claim is usually understood as one isolated sentence and claim detection is most often designed as a binary sentence classification task. But there are also exceptions. Arslan et al. [2] use a taxonomy with three classes, Konstantinovskiy et al. [9] have seven classes, and

[1] The code can be found on https://github.com/SamiNenno/Claim-Detection and the dataset is available on Gesis: https://search.gesis.org/research_data/SDN-10.7802-2538?doi=10.7802/2538.

Table 1 F: factuality and CW: checkworthiness

Corpus	Rows	Label	Best score	Source
Claimbuster [2]	23,533	CW*	F1 = .91	US-Presidential Debates
CT19 [12]	23,501	CW***	MAP = .17	US-Presidential Debates
CT20 [13]	962	CW**	MAP = .81	Tweets on COVID-19
CT21 [14]	45,619	CW***	MAP = .40	US-Presidential Debates
CT22 [15]	2891	CW**	F1 = .70	Tweets on COVID-19
TATHYA [16]	15,735	CW***	F1 = .26	US-Presidential Debates
Claimrank [10]	5415	CW***	MAP = .43	US-Presidential Debates
IndianClaims [17]	953	CW*	F1 = .70	Indian political debates
C/NC [9]	4777	F	F1 = .83	UK political TV-shows
Germeval [18]	4188	F	F1 = .76	Facebook Comments
LESA [19]	9981	F	F1 = .89	Tweets on COVID-19

CW*: checkworthiness is understood as "being of interest to the general public", CW**: "should be checked by a professional factchecker", and CW***: retrieved the label by matching actual factchecking articles

Gencheva et al. [10] model it as a rating task. However, often even these approaches are tackled in a binary format, too [11].

The most notable approach to claim detection is the *Claimbuster-project*[2] that contributed a widely used dataset [2] and built models that achieved strong results on it [11]. In their, and most other formulations, claim detection is the task of classifying *checkworthy* claims (Table 1). Checkworthiness is defined differently by different authors. In case of Arslan et al. [2] checkworthy claims are understood as "factual claims that the general public will be interested in learning about their veracity". Alam et al. [20] asked the annotators to label a sentence as checkworthy if the they could affirm the following question:" Do you think that a professional fact-checker should verify the claim in the tweet?" A third approach was taken by, for example, Shaar et al. [14], who matched sentences from US-presidential debates with factchecking articles by PolitiFact and labeled them as checkworthy if there was a corresponding article.

This and similar definitions have also attracted criticism. Allein and Moens [21] note that checkworthiness depends on the prior knowledge of a person, which makes the concept inherently subjective. They conclude that checkworthiness is an inapt concept to guide claim detection. One can add that the definition's reliance on "the general public" is already idealized as the public is often fragmented and different social groups differ in their values or political ideologies. Both aspects are relevant to determine if a claim should be fact-checked.

Arguing that checkworthiness is an editorial decision that is best left to human fact-checkers, Konstantinowskiy et al. [9] propose an alternative formulation of the claim detection task. They focus on factual claims, understood as sentences about

[2] https://idir.uta.edu/claimbuster/.

statistics, legal affairs, or causal relationships and opposed to sentences about personal experiences like "I woke up this morning at 7 a.m.". Risch et al. [18] and Wilms [22] created a German language dataset for the same task. However, in their taxonomy, a positive example is a claim to truth or a sentence that provides external evidence (links, quotes, etc.) rather than internal evidence (personal experience). Gupta et al. [19] define a claim as stating or asserting that something is the case, with or without providing evidence or proof. They use a similar scheme as the previous authors but they also consider personal experiences and humor/sarcasm as positive instances. All three approaches reach strong results (see Table 1).

Computational linguistics

A line of research in computational linguistics that shares many similarities to PCD is Dialogue Act classification (DA). One famous corpus is the Switchboard Dialogue Act corpus (SwDA) that was originally introduced by Jurafsky et al. [23] and provides a fine-grained labeling scheme of 43 classes that involve, for example, opinion-statements, non-opinion-statements, or different types of questions. The task is approached as text classification but also as sequence labeling and there have been attempts [24] that reach scores of 82.9 in F_1 (and 91.1 on the MRDA corpus [25]). DA and PCD follow the same theoretical tradition that is based on the works of Austin and Searle.

Argumentation Mining (AM) is closely related to claim detection as it aims at identifying the components of an argument and their relation. Similar to claim detection, extracting claims lies at the heart of AM. However, their definitions are not the same. In AM a claim is understood as a conclusion rather than a premise [e.g., [26]], while for claim detection both can be claims. Moreover, often sentences or textual spans that are understood as claims in AM would rather qualify as opinions from the perspective of claim detection.

The structure of propositional claim detection

Propositional content

PCD is about detecting sentences with propositional content. This term is used in philosophy and mathematical logic and denotes sentences that have a truth value, i.e., that can be true or false [27]. The concept of truth value is different from the concept of truth. Sentences that have a truth value are possibly but not necessarily true. They can also be false. Take the sentences "All cows are ruminants." and "Are all cows ruminants?" Even without knowing the word "ruminant", one can acknowledge that the first sentence can be true or false but the second cannot. Accordingly, we do not need to know if a sentence is true, in order to know that it has a truth value. Note that sentences with truth values are a broader class than factual sentences and the two sentence types should not be confused (see "Discussion" section).

Sentences that carry a truth value meet two minimal conditions: (1) the sentence must have a *condition of satisfaction* and (2) this condition must have a *word-to-world direction of fit*. For a sentence to have a condition of satisfaction is to have a relation to the world that can be fulfilled.[3] For example, saying that Japan has a total population of (roughly) 125 million relates to the world and it is satisfied because Japan does have about 125 million citizens *(a)*. Saying that Japan has the dirtiest streets worldwide also relates to the world but it is not satisfied because Japan's streets are extremely clean *(b)*. And if the Japanese digital minister pledges that the government will offer more digital services, his sentence relates to the world and it is satisfied if the government actually manages to provide more digital services *(c)*. However, other sentence types like apologies or congratulations cannot be satisfied. If the digital minister of Japan apologizes for not providing more digital services, his utterance cannot be satisfied. The purpose of his statement is neither referring to a desired future state of affairs nor is it a statement that can be true or false. It is meant as an apology, which is something different.

The second condition is a *word-to-world direction of fit*. This term dates back to the works of John Searle and is opposed to the *world-to-word direction of fit* (see [28, p. 100ff.]). Examples (a) and (b) share a word-to-world direction of fit. This means that sentences (a) and (b) must correctly represent the world in order to be satisfied. Sentence (a) does that while sentence (b) is a misrepresentation of the world. In such cases, we speak of propositional sentences, i.e., of sentences that carry a truth value. Opposed to this are sentences like (c) that have a world-to-word direction of fit. What the digital minister pledges is not intended to correctly represent the world. Instead, the world is supposed to adapt to the words of the Japanese digital minister so that we can say that his promise is fulfilled.

PCD taxonomy

As the purpose of PCD is to detect sentences that carry a truth value, the task is to identify sentences that (1) have a condition of satisfaction and (2) a word-to-world direction of fit. For reasons that will be explained shortly, PCD also adds a third condition: (3) the sentence must be in the present or past tense. Sentences that meet conditions (1)–(3) are called `assertions`. They are sentences in the past or present tense that carry truth values. Sentences that only meet conditions (1) and (2) but are in the future tense are called `predictions`. Sentences that meet condition (1) but not (2) are called `opinions`. Opinions have a world-to-word direction of fit and can be put in any tense. Sentences that neither meet condition (1) nor (2) are called `other` (Table 2).

PCD's structure can be mapped to the (German) grammatical structure in large parts but not entirely. The most obvious correlation is tense as it is a grammatical concept. However, there is more. The German language knows five grammatical sentence types: declarative (statement), interrogative (question), imperative

[3] Note that "world" is not limited to the physical world in this context. Sentences about, for example, mental states or social constructions do also have a relation to the world.

Table 2 Taxonomy for PCD

	Assertion	Prediction	Opinion	Other
Description	Statement about the present or past	Statement about the future	Personal or group standpoint or point of view	Sentence that cannot be evaluated in terms of true or false
Condition of satisfaction	Yes	Yes	Yes	No
Word-to-world direction of fit	Yes	Yes	No	No
Grammatical structure	Declarative sentence, past or present tense	Declarative sentence, future tense	Declarative sentence, any tense	Non-declarative sentence, any tense
Indicators	Supporting, quote, "I know that", "This proves that", etc	Contains "will" for future tense	Requires support, modal verbs, "In my opinion", "I want", etc	Ends with "?" or "!" or misses a verb
Example 1	It is war in Europe	We will face the same challenge again in autumn as last autumn	However, the freedoms of all must be taken into account	8 units per head, totally crazy!
Example 2	"We have to build trust," says Karsten Sach, head of the German delegation	With this, we will not achieve the 1.5° target	We want an efficient state and a citizen-friendly administration	Coal phase-out, climate change, sector coupling: the briefing for the energy and climate sector

(command), exclamation (affect), and optative (wish) sentences (see, for example, [29]).[4] Sentences that have a condition of satisfaction are almost exclusively declarative sentences. Declarative sentences are sentences in which the verb comes at second position and the sentence ends with a period sign.

The reduction of propositional sentences to declaratives, however, is an imperfect one. An exception are rhetorical questions, that can carry propositional content but which do not qualify as declarative sentences. For example, if a Fox News moderator rhetorically asks if the 2021 US-election was stolen, it can often be understood as a statement that the election results are not legitimate. This shortcoming is, for the moment, accepted and must be subject to future work.

While the difference between `assertions` and `opinions` on the one side and `predictions` and `other` on the other side can be reduced entirely to grammatical criteria, namely tense and sentence type, this is not the case for the difference between "`assertions`" and "`opinions`". There is no purely grammatical criterion that exhaustively separates these two classes. In many cases, it is necessary to look at the actual meaning of a given sentence in order to classify it as either one of the two categories. I will briefly outline some heuristics to differentiate between the two classes. More details can be found in the codebook.

1. Within an argument, `assertions` often take the supporting role, while `opinions` require support, i.e., `assertions` are often used as justifications, while `opinions` need justification.
2. Sentences that express `opinions` often contain modal verbs like "should" or "must".
3. `Assertions` often contain quotes. Note that in this case, we evaluate if something is correctly quoted and not if the content of the quote is correct.
4. `Assertions` often start with "We see that", "I know that", "This proves that", while `opinions` often start with "I want that" or "It is our opinion that".

Finally, one last remark about why tense is included in PCD. Tense only matters for the distinction between `predictions` and the other classes. The reason why `predictions` form a category on their own is that PCD is designed to assist fact-checkers. Fact-checkers are usually not interested in `predictions` as statements about future events can often not be verified.[5] Statements about the future do not necessarily use the future tense but can also be in the present tense and use temporal indicators like "tomorrow". However, these indicators are dependent on the time of utterance. Given that PCD is an NLP task, this matters strongly. Often the time of utterance cannot be inferred from the sentence alone but requires contextual information or meta-data. In order to avoid this complication, PCD works with the limited understanding of `predictions` as declarative sentences in the future tense. Another advantage is that this reduces difference of `predictions` to other classes to difference in grammar (tense), which is a key motivation for PCD.

[4] Sentences with subordinate clauses are common to the German language. For sentence with multiple clauses that fall into different classes, the annotators were told to use the label for the main clause rather than the subordinate clauses.

[5] https://correctiv.org/faktencheck/faq-haeufig-gestellte-fragen-an-das-faktencheck-team/.

Data selection and annotation

The dataset for PCD consists of a diverse set of (German) sources: Newspaper articles (*Die Zeit*, *Tagesschau*, *Tagesspiegel*, *taz*, *Süddeutsche Zeitung*), political TV shows (*Anne Will* and *Hart aber Fair*), party manifestos,[6] and tweets.[7] This broad array of text sources is meant to cover spoken and written text on the one side and formal and casual writing styles on the other. The time period that is covered in the dataset ranges from 1994 to 2022 with a special focus on the years at which national elections took place.

There were 4 coders in total: three student assistants, who were paid according to the collective agreement for student employees, and the author. The training was followed by an evaluation in which all coders had to annotate the same 225 sentences. They achieved a Krippendorff's alpha of 0.72 and then started the actual annotation process. The entire coding process, including the training, lasted about three months, with weekly workloads of about 400–700 sentences. To increase the output, about one third of the sentences were annotated by only one coder each. In order to speed up the annotation process, there was a rule-based (`Regex`) pre-annotation for `predictions` and `other`. These classes could be filtered with high (but not full) accuracy and were double-checked by a human coder. The remainder of the sentences was coded by at least two coders each. The entire process resulted in a total of 8425 annotated sentences. For more information on the annotation process, see "Appendix A".

For the optimal use of the limited resources, a pool-based Active Learning (Active Learning, for short) approach was chosen for selecting the data for annotation (Settles, 2010). The core assumption for Active Learning is that during the annotation process, the quantity of possible annotations is limited (due to financial and/or time constraints) but there is (almost) unlimited access to unlabeled data. The unlabeled data is called the *pool*. One way to choose the data for annotation is to randomly sample from the pool. In Active Learning, samples from the pool are chosen in multiple rounds and according to a "smart" method. There are various methods for doing that and they all have the purpose of choosing samples such that they are more informative for the model and it learns better and faster than if they were sampled randomly. The result is that less data is required for the model to achieve good results. The data was drawn from three pools of different text sources (Table 3). For more information, see "Appendix B".

[6] https://manifesto-project.wzb.eu/.

[7] https://zenodo.org/records/7670098.

Table 3 Data sampling with active learning

Round	Pool	No. sentences	News (%)	Protocol (%)	Talk show (%)	SoMe (%)	Manifesto (%)
1	1	1426	26	25	24	24	0
2	1	1207	31	30	27	11	0
3	1	942	25	34	0	41	0
4	2	1342	0	21	0	52	27
5	3	1795	31	42	0	(2)	25
Rule	All	1713	21	42	7	25	4

Each round a batch of sentences from one of three pools of different text sources was actively chosen. Due to a mix-up it was not possible to assign the sentences from social media in round 5 to their correct round. This is indicated by the brackets

Method

Consensus labels

It has been recognized that data quality is essential for machine learning and often fixing data issues improves model performance much more than model-centric approaches like hyperparameter tuning. One area of data-centric AI is label quality: there is research on how to find the best consensus label given multiple conficting annotations and there is research on reducing label noise, i.e., detecting mislabeled data points.

For the present study multiple techniques were evaluated in order to derive the final label for a given sentence: (1) strict majority vote, (2) soft majority vote, (3) Confident Learning [30], (4) CROWDLAB [31], (5) Confident Learning + CROWDLAB.

(1) and (2) are the most basic strategies. For the strict majority vote, only sentences that were seen by 2 or more annotators were included. In case of an agreement ≤ 0.5 the sentence was discarded. For the soft majority vote, no sentence was discarded and ties were dissolved hierarchically: `assertion > opinion > prediction > other`. Strategies (3)–(5) are rather sophisticated and based on more assumptions. They are briefly outlined, however, for a more thorough discussion, see "Appendix C" and "Appendix D" and the original papers.

Confident Learning is a method to detect mislabeled data points. Label errors can come in different shapes, for example, if a data point is assigned to the wrong class or if it belongs to multiple classes but is only assigned to one class. Confident Learning is a method to determine if a mismatch between a model prediction and a gold label is because the model failed or because the gold label is incorrect. A machine learning model or ensemble is leveraged to predict a probability for each label that indicates if the label is correct or not. For the present study, Confident Learning was used to remove all sentences whose gold labels were marked as incorrect.

CROWDLAB is a method to find consensus labels for multiple and possibly conflicting annotations. This method is based on two criteria. First, the annotator quality for each coder is computed. The annotator quality is understood as the level

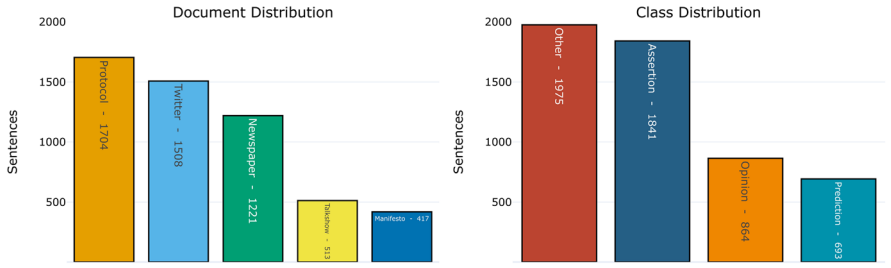

Fig. 1 Class and document distribution of final dataset

of agreement of one coder with the others. A coder who is often in line with the annotations by the others, receives a high quality score. Second, a machine learning model or ensemble is trained on the data and its predicted probabilities for each sentence are also taken into consideration for deriving the final label. Metaphorically speaking, the model simulates an additional annotator. By computing the annotator quality and simulating an additional annotator, it is possible to derive a label quality score even for sentences that were seen by only one annotator. For the present study, CROWDLAB was used to compute a quality score $q \in [0, 1]$ for each label so that labels with a score below a certain threshold (< 0.7) were removed from the dataset.

The detailed results for all of the mentioned methods can be found in "Appendix E". Methods (3)–(5) showed only little improvement compared to the strict majority vote and the soft majority vote performed relatively weak. For the final labels, strict majority vote was chosen, as it is a relatively simple method but performed well. The final dataset consists of 5373 sentences. The class and document distribution is displayed in Fig. 1. `Other` and `assertion` occur at a roughly equal amount and so do `opinion` and `prediction`. However, the first group makes up about 70% of the dataset, while the second group makes up the remaining 30%. The most frequent document type are protocols, followed by sentences that were drawn from Twitter.

Model selection

Six different embedding methods were used: BoW, Tf-idf, Word2Vec [32], GloVe [33], Fasttext [34], and Sentence Transformer [35]. The first two were calculated using the `scikit-learn` implementation with the default parameters. Pre-trained embeddings for Word2Vec and GloVe were retrieved from Deepset[8] and Fasttext embeddings were calculated using the `flair`-library [36]. All three were pre-trained on a Wikipedia corpus and their vocabulary is pre-defined. Since all three are word-level embeddings, they needed adjustment to sentence-level. Sentence embeddings were calculated by taking the mean of each word embedding of a sentence. Words that did not occur in the pre-defined vocabulary, were ignored.

[8] https://www.deepset.ai/german-word-embeddings.

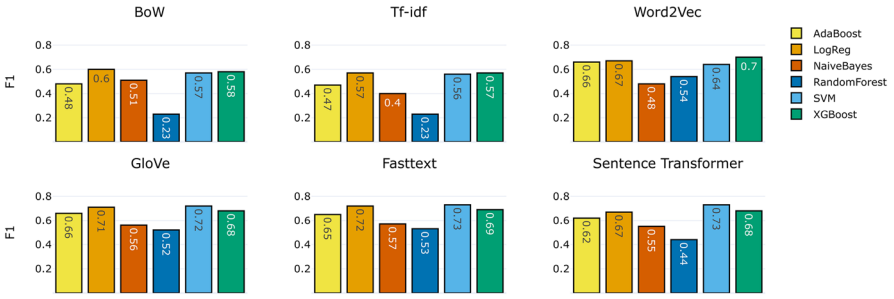

Fig. 2 Performance of classical machine learning models

Logistic Regression, Support Vector Machine (SVM), and Naive Bayes (Multinomial for BoW and Tf-idf and Gaussian for the other embeddings) were chosen as classical machine learning architectures. Ensemble Methods were included, too: Random Forest, AdaBoost, and XGBoost. All these models were implemented using `scikit-learn` with the default hyperparameter settings.

A set of transformer architectures were chosen as representatives of Deep Learning: The German base versions of Bert, Electra, Roberta[9] (called GottBert, see [37]) and DistilBert[10] as they can be found on Huggingface. It is expected that the transformer architecture, which is a driver for much of the recent progress in NLP, outperforms the classical architectures. However, transformer models require much computational resources and are more difficult to interpret than smaller models like SVM. For this reason, a broad selection of model architectures is evaluated.

The transformer-models were trained for 5 epochs and a batch size of 32 with the default hyperparameters implemented by the `transformers`-library (learning rate: 5e−5; weight decay: 0.0). Grid search over the number of epochs, batch size, learning rate, and weight decay was performed, however, none of the settings improved over the default settings.

Recall can be considered the most important metric for claim detection as it indicates how many of the relevant claims were actually found. However, high recall should not be achieved at the expense of precision because this means that the system also retrieves a lot of irrelevant claims. In this case, the difference does not matter strongly because for almost all experiments recall and precision are roughly equal. I therefore mostly report F_1 as the harmonic mean of precision and recall. All scores were obtained by averaging over the 10-fold stratified Cross-Validation.

[9] https://huggingface.co/uklfr/gottbert-base.

[10] https://huggingface.co/distilbert-base-german-cased.

Table 4 Performance of transformer models

Model	Accuracy	F_1	Recall	Precision
DistilBert	0.90	0.90	0.90	0.90
GottBert	0.88	0.88	0.88	0.88
Electra	0.91	0.91	0.91	0.91
Bert	0.91	0.91	0.91	0.91

Fig. 3 Confusion matrix and performance by document for Distilbert

Results

(Transformer-) models show strong Performance

Figure 2 displays the scores for the traditional machine learning models and for different embeddings. The scores reach from as low as 0.23-−0.72. There is no unambiguously best performing model. SVM shows the best performance in combination with GloVe, Fasttext, and transformer embeddings but not with the other embedding techniques. However, it turns out that Naive Bayes and Random Forests perform poorly for each embedding.

The results for the transformer architectures can be found in Table 4. As mentioned before, the performance remains constant across different metrics. The best performing transformer models are Bert and Electra with scores of 0.91 for all available metrics. DistilBert follows closely with a difference of 0.01, while GottBert performed worst of all transformer models. Nevertheless, the scores of the transformers are consequently higher than those of the classical machine learning models.

Figure 3 displays the performance of DistilBert, as it performed strongly in the previous tests and requires the least computational workload of all of the transformer models, depending on the individual class labels and document types. The highest class confusion is between `assertions` and `opinions`. The performance depending on the text type is balanced. There is no class for which the scores are much lower than for the others. Nevertheless, one can see that the model performance is weaker on sentences from manifestos and Twitter.

Table 5 Domain adaptation for Distilbert

Test domain	Accuracy	F_1	Recall	Precision
Vaccination	0.93	0.93	0.93	0.93
COP26	0.93	0.93	0.93	0.93
Turn of eras	0.93	0.93	0.93	0.93
Protocol	0.90	0.90	0.90	0.90
Twitter	0.83	0.82	0.83	0.82
Talkshow	0.91	0.91	0.91	0.92
Newspaper	0.88	0.89	0.88	0.89
Manifesto	0.87	0.87	0.87	0.88
20th century	0.90	0.90	0.90	0.90
21st century	0.80	0.79	0.80	0.80

Models adapt well to new domains

As mentioned in the introduction and discussed in the next section, one guiding assumption of this paper is that the reduction of class differences to grammatical differences (word order, punctuation, tense, etc.) leads to a strong generalization across domains. In order to test this hypothesis, several experiments were conducted on DistilBert.

For simulating new domains, the dataset was split according to different criteria (Table 5). To test generalization across different *topical domains*, all sentences that dealt with the "Turn of Eras"-speech by the German chancellor Olaf Scholz were separated and used as test set, while the remaining sentences were used as training set. The same was done for sentences about the German discussion about a mandatory vaccination during the COVID-19 epidemic and for sentences about the climate summit in Glasgow in 2021 (COP26). In order to test adaptation across *text sources*, the same procedure was conducted but the splitting criterion was if the sentences were drawn from parliament protocols, tweets, talkshow transcripts, newspaper articles, or party manifestos. It is expected that depending on the source, the texts have a rather monologic/dialogic or formal/informal character. Finally, the dataset was split according to the *time period* of the sentences. Some were drawn from the 20th century (1994–1998) and others from the 21st century (remainder). Note that for the sentences from the 20th century, data augmentation was performed due to scarcity.[11]

It can be observed that the performance remains high ($F_1 \geq 0.87$) and roughly constant across different domains. There are two exceptions. The model did not perform well on sentences from the 21st century. However, this is most likely because data from 1994 to 1998 is scarce and augmentation was only of limited help. The

[11] For data augmentation back-translation between German and English/Spanish was performed. For back-translation, text is translated to a different language and back to the original language. The assumption is that this process causes slight changes in the exact wording without changing the meaning of the sentence. The altered sentence are added to the data.

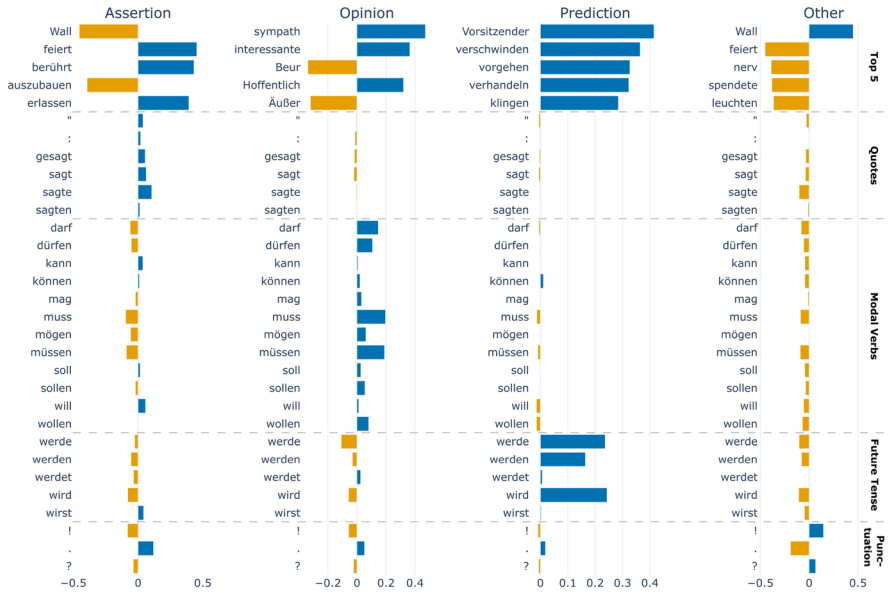

Fig. 4 SHAP-values for groups of class-relevant words

other exception is the relatively weak performance on Twitter data. However, the reduction to an F_1-score of 0.83 is still moderate, given the model did not see any social media data during the training phase.

Classifications are based on the intended criteria

It is argued that the strong domain adaptation is due to PCD's reduction of class differences to differences in grammar. The grammatical indicators for the different classes can be found in Table 2. For example, `assertions` often come in the form of quotes and are therefore marked by a colon and quotation marks. `Opinions` on the other side often contain modal verbs like *can*, *should*, or *must*. In order to find out if these indicators contribute to the classification by a PCD-model, an analysis using Shapley additive explanation (SHAP) values was conducted [38].

SHAP values are a measure of feature importance. SHAP values assign importance to features of individual examples, which leads to local explanations. Since PCD is for sentence classification, SHAP values indicate the importance of individual words or sub-words, depending on the embedding, for the classification of a given sentence into one of the four classes. For global explanations, i.e., explanations about the model's general behavior, it is possible to apply tendency measures, like the mean, on a sample of multiple examples. For the present study, SHAP values were computed for each sentence using 10-fold cross-validation.

Figure 4 shows the most important words for a given class for DistilBert, measured as the mean SHAP value of the given word with respect to each of the four

possible labels in the data set. The top 5 words are the most influential words for the given class. For `opinions` there are many subjective words (sympathy, interesting, hopefully), which aligns with the expectations. However, for the other classes it is difficult to make sense of the results. The next section includes conjugations of "to say", colons and quotation marks, which indicate quotes and which are strongest for `assertions`. The following section displays the importance of modal verbs and they are strongest for `opinions`. The fourth section displays the class-wise mean SHAP-values for "werden" (will) in different conjugations. Since this is the auxiliary verb for the future tense, it is the strongest for `predictions`. Finally, the class-wise mean SHAP-values for punctuations are displayed. As `other` consists mainly of questions and exclamations, we should expect question and exclamation marks to be of strong importance to this class. This is the case and, moreover, the period sign is very weak for `other` and strong for `assertions`. These results indicate that the model successfully picked up the criteria that were indicative for each class and which were guiding the human annotators. This might not come as a surprise but it shows that the model did not learn spurious correlations.

Discussion

PCD is more neutral and open than its alternatives

Claim detection is the task to identify claims that potentially carry misinformation. The practical goal is to provide a pre-selection of claims to fact-checkers in order to decrease their workload. Most approaches try to detect *checkworthy* claims, which is a notoriously vague term. In line with Konstantinovskiy et al. [9], PCD drops the notion of checkworthiness and focuses on claims to truth. However, PCD focuses on sentences with propositional content rather than on factual claims.

Propositional content is a broader notion than factuality. For example, a description of a personal experience like "I woke up this morning at 7 a.m." qualifies as a sentence with propositional content but not as a factual sentence according to the definitions by the other accounts. This is intentional and based on the assumption that factuality depends on the context. It might not be relevant from the perspective of fact-checking or misinformation at what time an average citizen woke up. But it might be relevant if the sentence was uttered by a politician. In this sense, PCD is a radicalization of factual claim detection because it makes even less assumptions about what is relevant for fact-checking and focuses exclusively on the fact that misinforming (as well as informative) sentences must carry truth values.

PCD shares a limitation with factual claim detection: it lacks a criterion for prioritizing one claim over another. Fact-checkers are not interested in simply *any* claim. Checkworthiness adds a notion of importance to the task and orders the claims according to their relevance. It was argued that checkworthiness is a problematic concept but nevertheless dropping it altogether leaves a gap that must be filled because it limits the applicability of claim detection.

The British factchecking organization *Full Fact* reports that by applying the models of Konstantinovskiy et al. [9] to parliament debates, Facebook posts, and Tweets, the output is roughly 80.000 claims per day.[12] Naturally, this is too much for manual analysis. For further filtering, they use heuristics. Certain topics like *Sports* or *Celebrities* are dropped and not forwarded to the factcheckers. In other words, focusing exclusively on claims to truth or factual claims is not enough. The models must be enriched by further selection criteria (a possible candidate is discussed in the last section). However, focusing exclusively on the fact that misinforming claims are sentences that can be true or false, PCD is strongly neutral and compatible to additions.

PCD shows strong performance across domains

It was shown that models can reach strong performance on the PCD-dataset. Even though classical machine learning models only reach poor or mediocre scores, transformer models achieve F_1-scores of 0.9 and more. This is also true for lightweight transformer models like DistilBert. In comparison, the strongest performance on the Claimbuster-dataset is 0.91 in F_1 [11], 0.76 for factual claim detection by [18] and 0.83 for [9]. The present results are also on the same level with approaches to Dialogue Act classification for which some approaches reach an F_1-score of 0.91 [24]. Accordingly, the question if PCD can reach competitive results can be answered positively.

The second guiding question of this paper was if PCD adapts well to new domains. It was assumed that PCD's focus on grammatical information, such as tense, word order, or punctuation, enhances generalization across domains. For example, most approaches to checkworthy claim detection use data that is drawn from US-presidential debates. It can be assumed that occurrences of checkworthy claims in this domain differ strongly from occurrences in other domains like COVID-19 or climate change with respect to their meaning and content. In contrast, grammatical cues like tense or word order are not affected by the topic. A declarative sentence is a declarative sentence no matter if the topic is COVID-19 or climate change.

In order to test this assumption, several experiments on domain adaptation were performed. The results showed positive results. This indicates that even though no training-dataset can exhaustively cover all possible domains, it is still likely that models for trained for PCD adapt well to previously unseen domains. This is because PCD focuses on relatively stable features like punctuation or modal verbs, as it was tested using SHAP-values.

[12] Personal communication.

Table 6 Examples for non-checkable claims to truth

Non-checkable Claim to Truth (NCCT)
He who does not seek, does not find
There is always hope
Since then, I only slide sideways on the motorway in my car, throw turtle shells and engage in littering with banana peels

Conclusion

This paper introduced *Propositional Claim Detection* (PCD) and a corresponding dataset. It further presented the results of extensive testing on this dataset. The two major limitations of PCD set the agenda for future research: It is a shortcoming that even though rhetorical questions can carry truth values, they are disregarded by PCD. Future research can build on the taxonomony of PCD and add rhetorical questions to it. Second, it was argued that checkworthiness is a problematic concept but dropping it altogether leaves a gap that must be filled. One possible candidate for it are *news values* [39]. News values or *newsworthiness* [40] shares similarities with checkworthiness but it enjoys a stronger theoretical and empirical grounding. Instead of relying on an abstract understanding of "what is interesting to the general public" scholars have found various concrete factors like proximity, timeliness, or conflict that make an event newsworthy. Furthermore, there have been studies that found that certain news values occur significantly stronger in misinformation than in "real news" [41, 42]. This can inform the automated detection of misinformation. Finally, there has been research on automatically detecting various news values that achieves strong results [43–45]. Future research should focus on combining PCD and news value detection. The result could be a classifier that aims at checkworthy claims to truth but avoids the aforementioned criticism of checkworthiness.

Despite these limitations, PCD is a solid foundation for claim detection: It is backed by a transparent taxonomy, achieves strong results and adapts well to new domains.

Appendix A: Annotation process

During the annotation process, annotators had difficulties to classify some sentence as either assertion or opinion. This problem arose especially for sentences that were tautological or strongly metaphorical (see Table 6). On the one side, they did not qualify as assertions as it was difficult to determine if they had propositional content, on the other side they did not match the intuitive meaning of what an opinion is. To tackle this problem, we introduced non-checkable claims to truth (NCCT) as a fifth category and redefined assertions as checkable claims to truth (CCT).

During the annotation process, it turned out that some classes are more similar to each other and hence more difficult to distinguish. The most problematic class

was NCCT. Originally, it was planned to merge the class with the opinion-class and keep it only for the sake of clarity during annotation. However, one of the biggest sources of error was that coders did not only disagree between NCCT and opinions but also between NCCT and CCT. Due to its similarity to both classes, NCCT could not simply be merged with one of them.

To solve the problem, there was a last round of annotation in which the coders were shown sentences that were labelled NCCT by others but this time there was no option to label the sentence as NCCT. Due to this, it was possible to merge many sentences that were labelled as NCCT by single coders, with either CCT or the opinion-class by performing a majority vote. All sentences that were labelled as NCCT by a majority of coders, were dropped for the machine learning experiments.

There was a number of sentence types which turned out to be a source of error. Vague descriptors: "Germany was a driver of the international process." What does it actually mean to be the driver in a process and can we check this? Implicit expressions of opinion: "With the CO2 price of all things, it is focusing on an instrument that will not solve the climatecrisis." It can be checked if carbon pricing is a successful method but this sentence has a condescending sound, which could also be an indicator for an opinion. General/Unspecific group descriptions: "For two years we all kept our distance, wore masks, minimised contacts." Who is "we" in this sentence and how could that be checked then? In case of doubt, the annotators were told to label CCT but the final decision was left to them.

In order to speed up the annotation process, some sentences were labeled automatically using a rule-based (`Regex`) approach. Even though it was not possible to filter all of the classes with full accuracy, the selection could at least be limited for some. Sentences belonging to `other` or to `prediction` were partly chosen with a rule-based strategy. Since `other` is partly constituted of questions and exclamations marked by "?" and "!", sentences containing these signs at the very end were filtered. In a second step one human annotator double-checked if these sentences were correctly filtered. For example, declarative sentences that quoted a question were discarded as they do not belong to `other`. The same procedure was conducted with sentences containing "werden" (will) and conjugations thereof, because this auxiliary verb marks the future tense. However, since not every sentence that contains this verb is a prediction, these sentences were also double-checked by a human annotator.

Appendix B: Active learning

For choosing the best sentences for annotation, Active Learning was used. One major strategy in Active Learning is to sample examples that the model is most uncertain about. The underlying assumption is that the model learns more from examples it is not certain about than from examples that it is certain about. Uncertainty of a model θ is expressed in its prediction probability for the given classes $P_\theta(\hat{y} \mid x)$. One Active Learning strategy is called margin sampling [46]:

$$x^* = \underset{x}{\operatorname{argmin}} P_\theta(\hat{y_1} \mid x) - P_\theta(\hat{y_2} \mid x)$$

where x^* is the most informative example and $\hat{y_1}$ and $\hat{y_2}$ are the first and second most likely classes. For example, if the model predicts `class A` with a probability of 0.8 and `class B` with a probability of 0.15 (the rest remaining 0.05 are spread among the other classes), the margin is 0.65. This is quite high because the model is confident in its prediction. If it is uncertain and predicts 0.4 and 0.3 instead, the margin is only 0.1. Active sampling in this version, means to compute the margin for all examples in the pool and then label the ones with the lowest margin.

However, margin sampling takes only the two most likely classes into account. The most popular Active Learning strategy, entropy sampling, on the other side, makes use of the probabilities for all classes:

$$x^* = \underset{x}{\operatorname{argmax}} - \sum_i P_\theta(\hat{y_i} \mid x) \log P_\theta(\hat{y_i} \mid x)$$

Entropy sampling is about finding sampling examples from the pool that have the largest entropy. The larger the entropy, the closer the distribution to uniform. A uniform distribution of the class probabilities means that the model finds all classes equally likely, which expresses a maximum degree of uncertainty.

For the present data set, entropy sampling seemed like a good strategy. However, one weakness of entropy (and margin) sampling is that it introduces a bias towards the model that is used to calculate the uncertainty. A transformer architecture and a logistic regression might have different prediction probabilities even if it is the same example and they were trained on the same data. Therefore, consensus entropy sampling was chosen. Instead of calculating the predictions probabilities on the examples from the pool for only one model, an entire ensemble of models is used. Entropy is then calculated on the mean of their prediction probabilities. The ensemble that was used consisted of a logistic regression, a support vector machine, and a XGBoost classifier as they come in the scikit-learn implementation. Furthermore, the ensemble consisted of two transformer architectures. One using the German base version of Bert,[13] the other using the German base version of Electra[14] as they can be found on Hugginface. There was no hyperparameter tuning for any model of the ensemble.

The data was drawn from three pools in five rounds and afterwards a sample of sentences was chosen from all three pools according to the rule-based strategy that was described before. The first pool consists of newspaper articles, German political TV talk shows, plenary protocols of the German parliament and tweets. All the data of the first pool is from the period between 2021 and 2022. The second pool consisted of tweets, political party manifestos, and plenary protocols of the German parliament. The time period for the second pool was 2017–2019. The last pool was

[13] https://huggingface.co/bert-base-german-cased.
[14] https://huggingface.co/deepset/gelectra-base.

drawn from newspaper articles, political party manifestos, and plenary protocols of the German parliament. The time period for the third pool was 1994–1998.

Appendix C: Confident learning

In the following, Confident Learning [30], i,e, the procedure to derive score and threshold for a given label are explained. First, the so-called *Confident Joint* is estimated.

$$\hat{\mathbf{X}}_{\tilde{y}=i,y^*=j} = \{\mathbf{x} \in \mathbf{X}_{\tilde{y}=i} : \hat{p}(\tilde{y}=j;\mathbf{x},\theta) \geq t_j\}$$

Where $\tilde{y}$ is a (possibly) erroneous label, which is commonly referred to as groundtruth label. y^* is a (unknown) true label that might or might not be equivalent to $\tilde{y}$. $\mathbf{x}$ is a training data example. θ is the model (or ensemble) that is used for the predictions. $\mathbf{x} \in \mathbf{X}_{\tilde{y}=i}$ is the set of training examples, whose groundtruth label $\tilde{y}$ equals i. $\hat{p}(\tilde{y}=j;\mathbf{x},\theta)$ is the predicted probability of model θ for example $\mathbf{x}$ to belong to class $\tilde{y}=j$, whereas j indicates the class with the maximum probability. t_j is a class-specific threshold that will be further specified shortly.

$\hat{\mathbf{X}}_{\tilde{y}=i,y^*=j}$ is read as: "The set of examples x whose groundtruth label $\tilde{y}$ equals i ($\tilde{y}=i$) but the true class label of x is more likely to be j ($y^*=j$)." And this is because the predicted probability for class j is above a certain threshold t_j. So the pressing question is is now: How is t_j estimated?

$$t_j = \frac{1}{|\mathbf{X}_{\tilde{y}=i}|} \sum_{\mathbf{x} \in \mathbf{X}_{\tilde{y}=i}} \hat{p}(\tilde{y}=j;\mathbf{x},\theta)$$

t_j is the mean predicted probability for class j for all examples x that have the groundtruth label $\tilde{y}=j$. An example: Assume for all sentences that have `Assertion` as groundtruth, the mean predicted probability for `Assertion` is 0.8. So $t_{Assertion} = 0.8$. Next, assume that for one sentence x with groundtruth `Opinion` the predicted probability for `Opinion` is 0.1 but for `Assertion` it is 0.82 (the rest is spread among the other classes). This means $\hat{p}(\tilde{y}=Opinion;\mathbf{x},\theta) = 0.1$ and $\hat{p}(\tilde{y}=Assertion;\mathbf{x},\theta) \geq t_{Assertion}$ is true. In such a case, we would say that x is falsely labelled as `Opinion` and its true label is `Assertion`.[15]

In sum, the score of an example x is the prediction probability for the most likely class and the threshold is t_j as it was described before. If x's predicted class j and its ground truth label i are not the same and the score of x is above threshold t_j, x is excluded for this experiment.

[15] If the prediction probability for `Assertion` was 0.7 and therefore below the threshold, we would not speak of an erroneous label (aleatic uncertainty) but of a erroneous prediction (epistemic uncertainty).

Table 7 Experiments with different methods to derive a final label

Experiment	No. sentences	Criteria
E0	5373	Strict majority vote, no confident learning
E1	8425	Soft majority vote, no confident learning
E2	7668	Soft majority vote, confident learning
E3	7314	CROWDLAB labels, no confident learning
E4	7064	CROWDLAB labels, confident learning

Fig. 5 F_1-scores for different models, different embeddings for each experiment

Appendix D: CROWDLAB

To derive labels with CROWDLAB [31], an ensemble of models is trained on the majority labels of the data set (the same ensemble that was used for Active Learning). The ensemble is treated as a new annotator. In this case as the fifth annotator. In a second step, a weighted ensemble of all five annotators determines a quality score $q \in [0, 1]$ for each label. The higher the score, the more likely the label. The weight for the ensemble is dependent on its accuracy on the majority labels. The weight of each annotator is dependent on their overall agreement with the other annotators (for the exact mathematical formula, see [31]). All labels with a quality score < 0.7 were discarded.

There are two advantages of CROWDLAB for the present study. First, due to the model as a fifth annotator, it was possible to let some sentences only be seen by one annotator and still not rely only on their judgment. This increased the amount

Fig. 6 Detailed scores. Line chart displays F_1-score for each document type and experiment

of sentence that could be annotated. Second, for the same reason, it was possible to break ties in a more reliable fashion than with a majority vote. This allowed us to have only two annotators for a sentence, even in danger of a possible tie.

Appendix E

In the following, the detailed results for the different experiments E0–E4 are presented (see Table 7). Figure 5 displays the (weighted) F_1-scores for different models and different embeddings on each experiment. All scores are obtained by averaging over the 10-fold stratified Cross-Validation. The two most striking observations are the high F_1-scores for the transformer architectures and that model performance follows the same pattern on the five experiments, independent of model or embedding. The highest score (0.94) was achieved by DistilBert, Bert, and Electra on E4, followed by GottBert (0.92). The second-best experiment is E0 (0.88-−0.91). Their F_1 -score for each experiment about 0.2-−0.3 points higher than the best F_1 of the non-Deep Learning models, irrespective of the embedding.

With regard to the more traditional models, SVM and Logistic Regression performed the best across the different embeddings. SVM achieved a F_1-score of 0.73 on Fasttext embeddings for E4. The two worst performing models are Naive Bayes and Random Forest. They are the bottom two for almost all embeddings.

The F_1-score for all models and embeddings follows, with some exceptions, the same pattern. E4 is the data set on which the models performed the best, followed by E3, and almost-tie between E2, E0, and performance for E1 is by far the worst. This indicates that Confident Learning has a positive impact on the models' performance. Especially, the leap from E1 to E2, which is about 0.1 in F_1 for the transformer models, is noteworthy.

CROWDLAB has an effect, too, even though it is less than that of Confident Learning and almost the same as strict majority vote. Also, CROWDLAB and Confident Learning do not seem to be mutually exclusive, as CROWDLAB label can

still benefit from Confident Learning. Also, models performed better on E0 than on E1. In other words, strict majority vote yielded better results than soft majority vote. This indicates that the labels that were controversial among the annotators, also confused the models during training.

Figure 6 displays the DistilBert scores with a stronger focus on the individual classes. The confusion matrices show that the major source of error for all experiments was the confusion of assertion and opinion. This is the strongest for E1, which confirms the hypothesis from the previous section: these classes cause the most confusion for humans and models alike. The line chart in Fig. 6 shows the F_1-Score of DistilBert in E4 relative to the document type. The performance on Twitter data was for all experiments the lowest. Given that it is the only social media source in the data set and that social media posts are often grammatically flawed, contain incomplete sentences, and use emojis, this is no surprise. What is a surprise on the other side is that sentences from talk shows achieved the best scores. Since sentences from talk shows have the least frequency, one might expect that the model would not perform best on them.

Acknowledgements We thank Anna Fischer, Christopher Richter, and Irina Kühnlein for their useful feedback and help with the annotations.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability The datasets is accessible on Gesis: https://search.gesis.org/research_data/SDN-10.7802-2538?doi=10.7802/2538.

Declarations

Conflict of interest There are no Conflict of interest.

References

1. Atteveldt, W., & Peng, T.-Q. (2018). When communication meets computation: Opportunities, challenges, and pitfalls in computational communication science. *Communication Methods and Measures, 12*(2–3), 81–92. https://doi.org/10.1080/19312458.2018.1458084
2. Arslan, F., Hassan, N., Li, C., & Tremayne, M. (2020). A benchmark dataset of check-worthy factual claims. In *Proceedings of the international AAAI conference on web and social media* (Vol. 14, pp. 821–829).
3. Welbers, K., Atteveldt, W. V., & Kleinnijenhuis, J. (2021). Extracting semantic relations using syntax: An R package for querying and reshaping dependency trees. *Computational Communication Research, 3*(2), 180–194.

4. Guo, Z., Schlichtkrull, M., & Vlachos, A. (2022). A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics, 10*, 178–206. https://doi.org/10.1162/tacl_a_00454
5. Ostrowski, W., Arora, A., Atanasova, P., & Augenstein, I. (2021). Multi-hop fact checking of political claims. In *Proceedings of the thirtieth international joint conference on artificial intelligence* (pp. 3892–3898). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2021/536. Accessed 21 September 2023.
6. Hardalov, M., Arora, A., Nakov, P., & Augenstein, I. (2022). A survey on stance detection for mis- and disinformation identification. In *Findings of the Association for Computational Linguistics: NAACL 2022* (pp. 1259–1277). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-naacl.94. Accessed 27 October 2023.
7. Glockner, M., Hou, Y., & Gurevych, I. (2022). Missing counter-evidence renders NLP fact-checking unrealistic for misinformation. In *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 5916–5936). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.397. Accessed 21 September 2023.
8. Nakov, P., Corney, D., Hasanain, M., Alam, F., Elsayed, T., Barrón-Cedeño, A., Papotti, P., Shaar, S., & Da San Martino, G. (2021). Automated fact-checking for assisting human fact-checkers. In *Proceedings of the thirtieth international joint conference on artificial intelligence* (pp. 4551–4558). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2021/619. Accessed 2 June 2022.
9. Konstantinovskiy, L., Price, O., Babakar, M., & Zubiaga, A. (2021). Toward automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection. *Digital Threats: Research and Practice, 2*(2), 1–16. https://doi.org/10.1145/3412869
10. Gencheva, P., Nakov, P., Màrquez, L., Barrón-Cedeño, A., & Koychev, I. (2017). A context-aware approach for detecting worth-checking claims in political debates. In *Proceedings of the international conference Recent Advances in Natural Language Processing, RANLP 2017* (pp. 267–276). INCOMA Ltd. https://doi.org/10.26615/978-954-452-049-6_037. Accessed 30 September 2021.
11. Meng, K., Jimenez, D., Arslan, F., Devasier, J. D., Obembe, D., & Li, C. (2020). Gradient-based adversarial training on transformer networks for detecting check-worthy factual claims. arXiv:2002.07725 [cs]. Accessed 18 November 2021.
12. Elsayed, T., Nakov, P., Barrón-Cedeño, A., Hasanain, M., Suwaileh, R., Da San Martino, G., & Atanasova, P. (2019). Overview of the CLEF-2019 CheckThat! Lab: Automatic identification and verification of claims. In F. Crestani, M. Braschler, J. Savoy, A. Rauber, H. Müller, D. E. Losada, G. Heinatz Bürki, L. Cappellato, & N. Ferro (Eds.), *Experimental IR meets multilinguality, multimodality, and interaction. Lecture notes in computer science* (pp. 301–321). Springer. https://doi.org/10.1007/978-3-030-28577-7_25
13. Shaar, S., Nikolov, A., Babulkov, N., Alam, F., Barron-Cedeno, A., Elsayed, T., Hasanain, M., Suwaileh, R., & Haouari, F. (2020). Overview of CheckThat! 2020 English: Automatic identification and verification of claims in social media.
14. Shaar, S., Hasanain, M., Hamdan, B., Ali, Z. S., Haouari, F., Nikolov, A., Kutlu, M., Kartal, Y. S., Alam, F., Beltrán, J., Elsayed, T., & Nakov, P. (2021). Overview of the CLEF-2021 CheckThat! Lab Task 1 on check-worthiness estimation in tweets and political debates. In *CLEF 2021—Conference and Labs of the Evaluation Forum*, September 21–24, 2021, Bucharest, Romania (p. 24).
15. Firoj, A., Dalvi, F., Shaar, S., Durrani, N., Mubarak, H., Nikolov, A., Martino, G. D. S., Abdelali, A., Sajjad, H., Darwish, K., & Nakov, P. (2021). COVID-19 Infodemic Twitter dataset. In *Proceedings of the fifteenth international AAAI Conference on Web and Social Media (ICWSM 2021). Harvard Dataverse*. https://doi.org/10.7910/DVN/XYK2UE. Accessed 23 August 2022.
16. Patwari, A., Goldwasser, D., & Bagchi, S. (2017). TATHYA: A multi-classifier system for detecting check-worthy statements in political debates. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. CIKM '17* (pp. 2259–2262). Association for Computing Machinery. https://doi.org/10.1145/3132847.3133150. Accessed 23 November 2021.
17. Jha, R., Motwani, E., Singhal, N., & Kaushal, R. (2023). Towards automated check-worthy sentence detection using Gated Recurrent Unit. *Neural Computing & Applications, 35*, 11337–11357. https://doi.org/10.1007/s00521-023-08300-x
18. Risch, J., Stoll, A., Wilms, L., & Wiegand, M. (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*

(pp. 1–12). Association for Computational Linguistics. https://aclanthology.org/2021.germeval-1.1. Accessed 7 October 2022.
19. Gupta, S., Singh, P., Sundriyal, M., Akhtar, M. S., & Chakraborty, T. (2021). LESA: Linguistic encapsulation and semantic amalgamation based generalised claim detection from online content. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume* (pp. 3178–3188). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.eacl-main.277. Accessed 7 December 2022.
20. Alam, F., Shaar, S., Dalvi, F., Sajjad, H., Nikolov, A., Mubarak, H., Da San Martino, G., Abdelali, A., Durrani, N., Darwish, K., Al-Homaid, A., Zaghouani, W., Caselli, T., Danoe, G., Stolk, F., Bruntink, B., & Nakov, P. (2021). Fighting the COVID-19 Infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society. In *Findings of the Association for Computational Linguistics: EMNLP 2021* (pp. 611–649). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.56. Accessed 22 August 2022.
21. Allein, L., & Moens, M.-F. (2020). Checkworthiness in automatic claim detection models: Definitions and analysis of datasets. In M. Duijn, M. Preuss, V. Spaiser, F. Takes, & S. Verberne (Eds.), *Disinformation in open online media. Lecture notes in computer science* (pp. 1–17). Springer. https://doi.org/10.1007/978-3-030-61841-4_1
22. Wilms, L., Heinbach, D., & Ziegele, M. (2021). Annotation guidelines for GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. *Excerpt of an unpublished codebook of the DEDIS research group at Heinrich-Heine-University Düsseldorf (full version available on request)*.
23. Jurafsky, D., Shriberg, L., & Biasca, D. (1997). Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13.
24. Raheja, V., & Tetreault, J. (2019). Dialogue act classification with context-aware self-attention. In J. Burstein, C. Doran, & T. Solorio (eds.), *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. https://doi.org/10.18653/v1/N19-1373. Accessed 6 November 2023.
25. Ang, J., Liu, Y., & Shriberg, E. (2005). Automatic dialog act segmentation and classification in multiparty meetings. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005* (Vol. 1, pp. 1061–10641). https://doi.org/10.1109/ICASSP.2005.1415300. ISSN: 2379-190X.
26. Stab, C., & Gurevych, I. (2014). Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 46–56). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1006. Accessed 21 September 2023.
27. McGrath, M., & Frank, D. (2020). Propositions. In E. N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy, Winter 2020 edn*. Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/win2020/entries/propositions/. Accessed 12 January 2023.
28. Searle, J. R. (1999). *Mind, language, and society: Philosophy in the real world*. New York: Basic Books.
29. Hentschel, E., & Weydt, H. (1994). *Handbuch der Deutschen Grammatik* (2 ed.). De Gruyter. https://dafdigital.de/ce/elke-hentschel-harald-weydt-handbuch-der-deutschen-grammatik/detail.html. Accessed 15 June 2022.
30. Northcutt, C., Jiang, L., & Chuang, I. (2021). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research, 70*, 1373–1411. https://doi.org/10.1613/jair.1.12125
31. Goh, H. W., Tkachenko, U., & Mueller, J. (2022). Utilizing supervised models to infer consensus labels and their quality from data with multiple annotators. https://doi.org/10.48550/arXiv.2210.06812. http://arxiv.org/abs/2210.06812 [cs, stat]. Accessed 19 October 2022.
32. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. https://doi.org/10.48550/arXiv.1301.3781. arXiv:1301.3781 [cs]. Accessed 6 January 2023.
33. Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162. Accessed 19 April 2021.

34. Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics, 5*, 135–146. https://doi.org/10.1162/tacl_a_00051
35. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 3980–3990). Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1410. Accessed 6 January 2023.
36. Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 annual conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)* (pp. 54–59).
37. Scheible, R., Thomczyk, F., Tippmann, P., Jaravine, V., & Boeker, M. (2020). GottBERT: a pure German language model. https://doi.org/10.48550/arXiv.2012.02110. arXiv. arXiv:2012.02110 [cs]. Accessed 16 January 2023.
38. Lundberg, S. M., & Lee, S.-I. (2017). 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.* (Vol. 30). Curran Associates Inc.
39. Galtung, J., & Ruge, M. H. (1965). The structure of foreign news: The presentation of the Congo, Cuba and Cyprus crises in four Norwegian newspapers. *Journal of Peace Research, 2*(1), 64–90. https://doi.org/10.1177/002234336500200104
40. Caple, H. (2018). News values and newsworthiness. In *Oxford research encyclopedia of communication*. Oxford University Press. https://doi.org/10.1093/acrefore/9780190228613.013.850. Accessed 16 November 2022.
41. Tandoc, E. C., Thomas, R. J., & Bishop, L. (2021). What is (fake) news? Analyzing news values (and more) in fake stories. *Media and Communication, 9*(1), 110–119.
42. Chen, X., Pennycook, G., & Rand, D. (2023). What makes news sharable on social media? *Journal of Quantitative Description: Digital Media*. https://doi.org/10.51685/jqd.2023.007
43. Potts, A., Bednarek, M., & Caple, H. (2015). How can computer-based methods help researchers to investigate news values in large datasets? A corpus linguistic study of the construction of newsworthiness in the reporting on Hurricane Katrina. *Discourse & Communication, 9*(2), 149–172. https://doi.org/10.1177/1750481314568548.
44. Bednarek, M., Caple, H., & Huan, C. (2021). Computer-based analysis of news values: A case study on national day reporting. *Journalism Studies, 22*(6), 702–722. https://doi.org/10.1080/1461670X.2020.1807393
45. Piotrkowicz, A., Dimitrova, V., & Markert, K. (2017). Automatic extraction of news values from headline text. In *Proceedings of the student research workshop at the 15th conference of the European Chapter of the Association for Computational Linguistics* (pp. 64–74). Association for Computational Linguistics. https://doi.org/10.18653/v1/E17-4007. Accessed 16 January 2022.
46. Settles, B. (2009). *Active learning literature survey*. Computer Sciences Technical Report 1648. University of Wisconsin-Madison.