Suspicious Sentence Detection and Claim Verification in the COVID-19 Domain

Elitsa Pankovska[1], Konstantin Schulz[2] and Georg Rehm[2]

[1]TU Berlin, Berlin, Germany
[2]DFKI GmbH, Berlin, Germany

Abstract

The processing, identification and fact checking of online information has received a lot of attention recently. One of the challenges is that scandalous or "blown up" news tend to become viral, even when coming from unreliable sources. Particularly during a global pandemic, it is crucial to find efficient ways of determining the credibility of information. Fact-checking initiatives such as Snopes, FactCheck.org etc., perform manual claim validation but they are unable to cover all suspicious claims that can be found online – they focus mainly on the ones that have gone viral. Similarly, for the general user it is also impossible to fact-check every single statement on a specific topic. While a lot of research has been carried out in both claim verification and fact-check-worthiness, little work has been done so far on the detection and extraction of dubious claims, combined with fact-checking them using external knowledge bases, especially in the COVID-19 domain. Our approach involves a two-step claim verification procedure consisting of a fake news detection task in the form of binary sequence classification and fact-checking using the Google Fact Check Tools. We primarily work on medium-sized documents in the English language. Our prototype is able to recognize, on a higher level, the nature of fake news, even hidden in a text that seems credible at first glance. This way we can alert the reader that a document contains suspicious statements, even if no already validated similar claims exist. For more popular claims, however, multiple results are found and displayed. We manage to achieve an F_1 score of 98.03% and an accuracy of 98.1% in the binary fake news detection task using a fine-tuned DistilBERT model.

Keywords

Sentence classification, Fact checking, Language models, COVID-19

1. Introduction

With the help of the internet and social media, getting information on practically any topic has become a rather simple task. However, the sheer amount of online content makes it impossible to have any control over its quality and reliability. One of the internet's most significant advantages, the freedom of speech, has turned into one of its most crucial challenges: anyone can write anything about any topic without having to take any responsibility for the impact their content may have on readers. False information exist in various ways, including misinformation and disinformation. A person sharing misinformation was misinformed themselves (i. e.,

ROMCIR 2022: The 2nd Workshop on Reducing Online Misinformation through Credible Information Retrieval, held as part of ECIR 2022: the 44th European Conference on Information Retrieval, April 10-14, 2022, Stavanger, Norway
elitsa.pankovska@gmail.com (E. Pankovska); konstantin.schulz@dfki.de (K. Schulz); georg.rehm@dfki.de (G. Rehm)
0000-0002-3261-9735 (K. Schulz); 0000-0002-7800-1893 (G. Rehm)

they believe what they are saying is true). In the case of disinformation, however, someone is spreading false information with the intention of misleading others. Online misinformation, often compared to online propaganda, has become a major problem. Many people tend to rely on online sources to get information on current events or topics without keeping in mind that not everything read online is objective. On top of this development, traditional media often make use of or republish online content without fact-checking its origin.

This development has become even more concerning with the COVID-19 pandemic. Due to the uncertainty that comes with the disease, it has become easier than ever to spread false claims and mislead large groups of people. While claims made by prominent public figures are usually validated swiftly, it is impossible to fact-check arbitrary statements on any topic made online. Manual fact-checking is nowadays performed by more than 300 official initiatives[1], but they still cannot cover everything. For this reason, it is crucial to find a more efficient way of performing fact-checking or at least to recognize possibly false news hidden in sources that appear to be trustworthy at first glance. On the other side, it is inefficient to fact-check every single sentence in an article, especially if it does not contain potentially suspicious claims. Hence, a two-step approach of estimating whether a sentence needs to be fact-checked, followed by its validation, can be a step towards automated fact-checking of whole text documents.

Our goal is the development of a tool that first attempts the identification of dubious sentences in small- to medium-sized English language documents on the topic of COVID-19, followed by a second step of fact-checking the claim, i. e., running the claim against external knowledge bases. We focus on the detection of false news based solely on the text of the content. False news detection is approached as a binary classification task. Claim validation is performed using a trusted knowledge base. This subsequent step covers a broader spectrum by ordering claims on a range from at least 1-5, instead of merely as "true" or "false". This work can help users to determine the credibility of online information and it can increase the productivity of fact-checkers by sparing them the work of validating claims, which were already fact-checked.

Section 2 defines key terms and provides an overview of the state of the art. Section 3 describes the methodology and decompose the challenge into individual steps. Section 4 describes the implementation and Section 5 explains the experimental setup and data sets. Section 6 provides an evaluation of our approach and Section 7 concludes the paper.

2. Related Work

Below we define key terms (Section 2.1), review more general previous research (Section 2.2) as well as related work concentrating on the COVID-19 domain (Section 2.3).

2.1. Topic-specific Definitions

According to the Cambridge English dictionary, *fake news* can be defined as "false stories that appear to be news, spread on the internet or using other media, usually created to influence po-

[1] https://reporterslab.org/fact-checking-count-tops-300-for-the-first-time/

litical views or as a joke".[2] Fake news can be further separated into different categories. Wardle [1] defines seven types of mis- and disinformation, the "mildest" case being (harmless) satire or parody. Further on the scale are false connection, misleading content, false context, and imposter content. The final two categories are manipulated and fabricated content. Both have the intention to deceive, with the latter one being completely false. As Wardle states, people tend to share content on social media without double-checking, so even the more "harmless" types of fake news spread quickly and have the potential to fool. We consider fake news to be text sequences of all types mentioned above. A *suspicious claim or statement* contains possibly false, or misleading information, or in general, is proved not to be entirely true. *Fact checking* the veracity of claims, statements or sentences in a text (document) includes the assessment of their average veracity using the ratings of existing relevant claims on a scale of 1 to 5.

2.2. State of the Art

In the following, we review the current state of the art in the area of automated fake news detection and automated claim verification.

2.2.1. Fake News Detection

There are three types of features used to detect purely textual fake news: content features, propagation features, and source features [2, 3]. Linguistic features, both lexical and syntactic, fall under the umbrella of content features, see, e. g., [4]. In contrast, approaches focusing on the source/user features analyze a source's reliability, previous behavior, etc. Propagation-based research relies on the assumption that fake news has a different propagation process, including that fake news spreads faster and reaches more people [5]. Current approaches [6, 7, 8] work on combining the different groups of features, the relations and connections between them [9] or crowd-sourced information to achieve better results, see, e. g., [10, 11].

Collins et al. [12] give an overview of the type of fake news and the different detection techniques (human fact-checkers, crowd-sourcing, ML, NLP, combinations thereof). However, they mostly cover outdated approaches, i. e., Gradient Boosted Decision Trees and Naive Bayes, and do not go in-depth with state-of-the-art ML algorithms and models. Vishwakarma and Jain [3], Li et al. [13], Gundapu and Mamidi [14] show that pre-trained language models (such as BERT) achieve better performance in fake news detection than traditional NLP algorithms, e. g., Support Vector Machines, Random Forests, or Naive Bayes. They can detect stylistic differences much better due to their learned understanding of language, achieved during pre-training [2]. Fact-checking initiatives, such as Fullfact, also strive towards automating their workflows by identifying fact-check-worthy claims using their own fine-tuned BERT model.[3]

For fake news detection, most state-of-the-art approaches focus on the article level and use groups of features. For full articles, the author feature is important but we do not have access to it in our experiments. Moreover, our goal is to detect fake news on a sentence level by just using content features. A major challenge for fake news detection on a sentence level is that widely accepted benchmark datasets do not exist. Existing datasets, such as CREDBANK

[2] https://dictionary.cambridge.org/dictionary/english/fake-news
[3] https://fullfact.org/about/automated/

[15], LIAR (created from fact-checks from Politifact) [16] and FakeNewsNet [17], although suitable for general fake news detection, are outdated and do not include data from the past year and consequently, do not prove useful for our use case. Moreover, they traditionally either include full articles or social media posts (most often tweets) and rely on crowd-sourcing. The tool that is most similar to our approach (it identifies claims, skips over irrelevant statements, and fact-checks individual claims), is ClaimBuster.[4] It relies on human data-labeling contributions and has been trained using a large dataset of sentences from political debates in the US from 1960-2016 [18]. They experiment with different "claim spotting" models, including SVM, bi-directional LSTMs, and Transformers but "claim spotting" and "claim checking" are two separate tools; plus, "claim spotting" is not yet optimized for COVID-19 related texts.

2.2.2. Automated Claim Verification

Verifying claims is a time-consuming process, which relies on manual research and information assessment. Previous work in this area is rather limited. The claim verification process cannot be automated fully when it comes to more than a simple information lookup. Still, knowledge bases, through which fact-checkers can at least find already validated claims, are a big step in the right direction. Chernyavskiy et al. [19] propose a system, which finds relevant Wikipedia articles, extracts relevant sentences, and determines if they support or refute a given claim. Other approaches use ClaimReview markup, which allows for easy tagging of fact-checks. It is used by all fact-checking initiatives, making it possible to assemble a collaborative knowledge base, accessible through the Google Fact Check Tools API (see Section 3.2). Other approaches include evidence retrieval using existing datasets, which can, similarly to fake news detection, be done using Transformer models [20]. Thorne et al. [21] use TF-IDF features to find the k-nearest documents and select sentences using TF-IDF similarity. Such implementations, however, cannot be a long-term solution since datasets need to be updated and models trained constantly. A considerable contribution towards COVID-19 knowledge retrieval are knowledge graphs (KGs), which use, e. g., the CORD-19 dataset. Wise et al. [22] present a KG with different relations between scientific articles on COVID-19. Domingo-Fernández et al. [23] provide an even more complex COVID-19 KG covering ten entity and 9484 relation types. Similar KGs can facilitate verified information retrieval for claim validation.

2.3. Approaches in the COVID-19 Domain

In terms of applying such approaches to the COVID-19 domain, there are already a number of published results. Manual claim validation is constantly being performed by fact-checking initiatives, e. g., Snopes, Politifact, etc. Below, however, we only discuss automatic approaches.

Wadden et al. [24] concentrate on scientific claims, they use CORD-19 as a source of evidence. Given a claim and CORD-19 abstracts, they identify abstracts that support or refute the claim. Instead of scientific articles, we focus on more general claims including, say, statements from politicians or social media captions, as long as they fall under the umbrella of COVID-19.

Vijjali et al. [25] propose a model, which takes a claim and explanations and determines the probability of the claim being in entailment with the explanations. A first model fetches

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

relevant explanations and is trained on sentence entailment. A second model identifies the veracity of a claim. Both models are trained on sequence classification and evaluated with Transformer architectures [26, 27, 28]. The approach shows the dominance of Transformer models, but the final model is not applicable in real-life scenarios because it cannot distinguish fake news from genuine news but merely considers explanations.

Barrón-Cedeño et al. [29] give an overview of the third edition of the CheckThat! lab, which was hold under the umbrella of CLEF 2020. The first task focuses on the importance of estimating the fact-check-worthiness of a claim. The dataset consists of COVID-19-related tweets, annotated manually into two classes (fact-check-worthy or not), based on the answers of two questions [29]. The best solutions use Transformer models (RoBERTa, BERT).

Das et al. [30] research the impact of using an ensemble approach rather than a single model for detecting COVID-19 related fake news. The results show that an ensemble of Transformer models, combined with soft voting (averaging out the prediction probabilities of multiple models for each class), delivers state-of-the-art results.

3. Methodology

The brief overview of the architecture we developed includes descriptions of the process pipeline and the tools and models used (Section 3.1). Section 3.2 presents the claim validation and the knowledge bases we used.

3.1. Process Pipeline

We process text in multiple steps. First, it is parsed into sentences, where each one is classified into one of two categories: a sentence, which contains possibly dubious claims related to COVID-19, or a regular one. For all regular sentences, the process ends here. For the remaining ones, we perform two additional steps: claim extraction and claim validation. The first one is done with an existing tool, while the latter is done by searching for similar claims in a trusted knowledge base. Finally, all results returned are analyzed and classified into six categories.

The first step of our fact-checking task is also our main focus, i. e., training a sentence classification model, which can detect sentences containing potentially suspicious claims or fake news. We experimented with a number of techniques including different types of Transformer-based language models [31], which exhibit state-of-the-art performance in multiple NLP tasks including BERT [26], RoBERTa [32], DistilBERT [33] and SciBERT [34], which was trained on a random sample of 1.14 million papers from Semantic Scholar. SciBERT outperforms BERT on tasks in the scientific domain, which our topic also partially overlaps with. Wadden et al. [24] report SciBERT to be among the best performing models when it comes to experiments using claims concerning COVID-19.

The subsequent step of sentence parsing and claim extraction is performed using spaCy, which supports tokenization, part-of-speech-tagging, etc.[5] We opt for a rather simple approach when it comes to claim extraction. Various processors can be added to the default spaCy pipeline, including a sentencizer to split text into sentences, which is the first step in

[5] https://spacy.io

our process pipeline. Punctuation removal can be performed using POS tagging. Stopword removal is supported out of the box through its set of stopwords, which can also be manually modified. In addition to spaCy, we also conducted experiments with Stanza [35].

3.2. Claim Verification

Automatic claim verification can be done in various ways, but mainly by searching for similar claims, which are already validated. This search can be done by using external knowledge graphs or knowledge bases or by using a dataset of validated claims. This means that the claim verification always relies on third-party data or services. We choose the Google Fact Check Tools for this step due to their reliability and up-to-date information.

ClaimReview is a type of structured markup, based on Schema.org, that fact-checkers can use to enrich their articles for search engines and social media platforms.[6] Each ClaimReview object includes multiple properties. Of interest to us is the reviewRating property, which is of type Rating[7] with a ratingValue that can be textual or numerical. The convention, however, is a numerical value between 1 and the number stated as "bestRating". In the case of ClaimReview, a value of 1 is equivalent to "False", while 5 means "True".

Google's Fact Check Tools have three main functions. The Google Fact Check Explorer is like the traditional Google Search, with the difference that a user searches for fact checks (validated claims). Results have a claimant, text of the claim and data about the review, including a textual rating. Second, users can add ClaimReview Markup to their site using the Markup Tool in order for their reviews to pop up as results in, e. g., the Fact Check Explorer. Of interest to us is the Claim Search API, which developers can use via HTTP GET requests to query the same set of Fact Check results available via the Fact Check Explorer.[8] The request body includes the query text (i. e., the claim), language code, pageSize and maxAgeDays. By limiting the maximum age of a claim review, we somewhat prevent fetching already outdated results. This is crucial especially in the COVID-19 domain, where the scientific community has repeatedly changed its conclusions about the effectiveness of certain protective measures, the effects/diffusion of COVID-19, etc.

The response body includes an array of fact-checked claims, where each one includes the claim text, the claimant, a claim date and the claim review with a textual rating. One crucial disadvantage of this JSON structure concerns the fact that there is only a textual rating provided rather than a numerical one, i. e., the textual ratings from multiple fact-checking sources cannot be compared, which calls for a meaningful mapping. With a numerical rating, we can easily compute an average from all relevant results, in order to have the final rating of our initial claim.

[6] https://schema.org/ClaimReview
[7] https://schema.org/Rating
[8] https://toolbox.google.com/factcheck/apis

4. Implementation

Below we present the architecture of our implementation and describe the individual steps.[9] Although our use-case addresses the COVID-19 domain, we believe that the following implementation is also applicable to other areas, such as political disinformation, climate change disinformation, etc.

4.1. Overall Architecture

The system takes a text as input and performs various processing steps. First, it is parsed into sentences using the spaCy library. Second, each sentence is classified as either suspicious or regular using a pre-trained Transformer model, fine-tuned using our dataset. Third, for all dubious sentences we perform claim extraction by simplifying the sentence using simple preprocessing, again using spaCy. Fourth, the simplified sentence is used as input for the Google Claim Search API. Fifth, from its results, textual ratings are mapped to numerical ones, and an average is taken to determine the veracity of the claim. The process is visualized using the Streamlit library and a custom Vue frontend component that presents the text to the user, with all suspicious sentences highlighted in colours corresponding to their veracity.

4.2. Fake News Detection

For all Transformer models (BERT, DistilBERT, SciBERT, RoBERTa) we use a standard HuggingFace implementation,[10] also making use of Auto Classes.[11] We tokenize sentences, with padding and truncation enabled and a maximum sequence length of 512, with the main training arguments being two epochs, train batch size 8, eval batch size 64, 2000 steps.

Additionally, we calculate the following metrics in each evaluation step: accuracy, cross-entropy-loss, precision, and recall, as defined in the sklearn.metrics module.[12] Finally, we use the Trainer class, which takes our training arguments, the train and validation datasets, the model initializing function, and our compute_metrics function. Once initialized, we run the train function to complete the fine-tuning procedure. The exact parameters, training time, evaluation results, etc. for all fine-tuning procedures are recorded in our W&B projects.

4.3. Claim Extraction

For the claim extraction step we simplify the initial sentence. We decided against extracting claims on the discourse level and focus upon individual sentences. First, each suspicious sentence is tokenized and all tokens of type "punctuation" are removed. Second, we remove all stop words from the sentence using spaCy's default set of stopwords. This results in an array of keywords (i. e., the claim), which we use to query Google's Fact Check Tool.

[9] https://github.com/elip06/covid19-fact-checking provides the full code of our implementation.
[10] https://huggingface.co/transformers/custom_datasets.html
[11] https://huggingface.co/transformers/model_doc/auto.html
[12] https://scikit-learn.org/stable/modules/model_evaluation.html

4.4. Claim Verification

We limit the results from the Google Fact Check Tool API to the English language only. An example query may look like this:

```
GET https://factchecktools.googleapis.com/v1alpha1/claims:search? language-
↪ Code=en&maxAgeDays=200&query=ginger%20cures%20corona&key=[YOUR_API_KEY]
```

To determine the veracity of a claim, we need to map the heterogeneous textual ratings of a claim to numerical ones since the API returns query results from *different* sources[13] that use their own way of expressing or encoding the result of the verification step. We map as follows:

- 1: “false”, “four pinocchios”, “inaccurate”, “miscaptioned”, “misattributed”, “scam”
- 2: “mostly false”, “three pinocchios”, “misleading”
- 3: “mixture”, “two pinocchios”, “biased”, “cherry-picking”, “not the whole story”, “exaggerates”
- 4: “mostly true”, “half true”, “one pinocchio”
- 5: “true”, “accurate”, “unbiased”, “correct”

With this approach, we combine the ratings of all relevant claims to calculate an average value and determine the veracity of the original claim, which is why we put some terms that are not exactly equivalent under the same umbrella (e. g., “true” and “unbiased”). It must be noted, though, that there is no standard or convention with regard to this mapping approach.

We use the default maximum number of ten results for each search query since the further one goes through the list of results, the more irrelevant and outdated they become (similarly to the standard Google Search). Additionally, the results for many of the claims are repetitive since the most prominent fact-checking initiatives fact-check the same claims.

We realise that this manner of computing numerical ratings is not perfectly accurate. The required fields for Google’s Fact Check products are only the fact checked article, the claim reviewed, and a rating text. While there is the option to also add a numerical rating, also indicating the best and worst possible ratings, they are present neither in the Fact Check Explorer nor in the Fact Check Tools API.

4.5. Deployment and Visualization

We implement an additional UI, which has two main functions: firstly, we make use of it for the qualitative evaluation of both steps of the system, by feeding text documents and looking at the final result. Moreover, the tool can be used by fact-checkers with no technical knowledge to easily check the credibility of texts and find suspicious claims and whether they have already been verified. For the UI prototype of our approach we use the Streamlit[14] framework so that we can develop the application in Python, also making use of Vue and Vuetify.[15]

The user can either upload a Word or plain text file or type text into a text field. The text is parsed into sentences, which are then classified. Once the claim extraction and queries are completed, the UI receives the following data as input: an array of all sentences and an array of

[13] Including Snopes, Washington Post, Health Feedback etc.

[14] https://streamlit.io

[15] https://vuejs.org, https://vuetifyjs.com

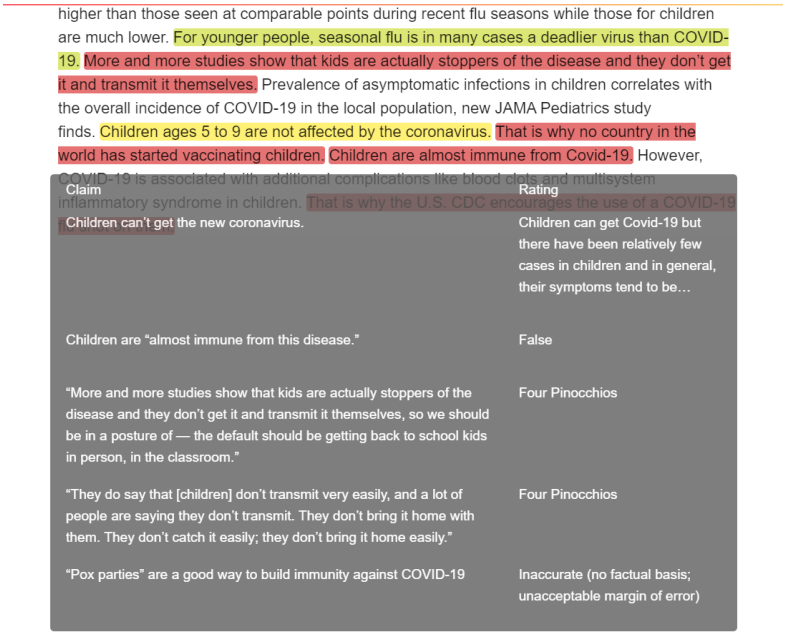


Figure 1: A processed text in the UI

labels/claims. The UI shows the text and an explaination of what the different highlight colours mean. When the user hovers over a suspicious sentence, for which relevant claims exist, those claims and their ratings are displayed (Figure 1).

5. Experimental Setup

5.1. Datasets

Many different datasets and publications exist for the wider COVID-19 domain. The primary purpose of the data is to fine-tune the pre-trained language models for the sequence classification but a dataset of whole documents in which individual sentences are labeled as "suspicious" does not exist, which is why we combine different datasets to achieve this structure. Initially, we used suspicious sentences taken from the archives of fact-checking websites, such as Snopes and Politifact, and regular ones from scholarly articles on COVID-19. While this dataset performed well during cross-validation, it showed weaknesses in real-life scenarios by labeling a significant amount of regular sentences as suspicious. Consequently, we use an additional dataset to increase the model's accuracy. A brief description of all used datasets follows. The first three are part of the initial dataset, on which multiple Transformer models are fine-tuned and compared. The last one is added later and used for fine-tuning all models one more time.

1. *COVID-19 Open Research Dataset (CORD-19)* consists of more than 500,000 scholarly articles, including over 200,000 with full text, about COVID-19 and related coronaviruses [36]. We use only a small portion, i. e., the abstracts of the most recent articles at the

time of data collection (the first week of January 2021). We use CORD-19 as the source for non-suspicious sentences.

2. *FakeCovid: A Multilingual Cross-domain Fact Check News Dataset for COVID-19* consists of articles in 40 languages but the majority have titles in English [37]. These articles have been fact-checked by reliable online sources. We use FakeCovid as the source for suspicious sentences in the initial dataset.
3. *CoAID (Covid-19 heAlthcare mIsinformation Dataset)* includes fake news on websites and social media, along with users' social engagement about such news [38]. We only use part of it since many of the claims are also present in FakeCovid.
4. *COVID19 Fake News Detection in English* consists of real and fake news on COVID-19 [30]. We improve our models using this dataset as it introduces a different type of regular sentences. Unlike the ones from scientific articles, the non-suspicious sentences are merely real news on the topic of COVID-19, which do not necessarily have the same structure or use scientific terms as the other ones do but are still just as trustworthy, i. e., the models can learn the specifics of fake news, rather than the differences between a scientific and non-scientific text.

All datasets are preprocessed in accordance with their specific structure and parsed into sentences using Stanza. We also performed additional cleaning (all sentences which either contain no verb or consist of less than four words are removed). The two sets of genuine and dubious sentences contain 7908 samples each. We merge together the two sets and split them into train (80%), and test sets (20%) using Scikit-learn's train_test_split function. The fourth dataset was added due to the unsatisfactory initial results, arriving at a final train dataset of 9426 (49.4%) suspicious samples and 9646 (50.6%) non-suspicious ones as well as a final test dataset of 2562 (48.3%) suspicious samples and 2742 (51.7%) non-suspicious ones.

5.2. Hyperparameter Search

The choice of hyperparameters can have a major influence on the machine learning models' performance [39]. For this reason, it is crucial to use a scalable hyperparameter optimization framework. Our choice is Ray Tune [40], due to its possibility of early termination of bad runs and intelligent choice of parameters within the defined space. Our search for hyperparameters is documented in a Weights & Biases project.[16] Experiments are conducted using the following hyperparameters:

- Learning Rate: between 1e-5 and 1e-3
- Number of Training Epochs: between 1 and 4
- Seed: between 1 and 42
- Per Device Train Batch Size: 8 or 16
- Warmup Steps: between 0 and 1000
- Weight Decay: between 1e-6 and 0.1

Most of the hyperparameters are an obvious choice, except for the random seed. However, Dodge et al. [41] found that by checking many different random seeds they obtained significant

[16] https://wandb.ai/elip06/distilbert-hyperparameter-tuning

improvements on the performance of BERT and some seeds are statistically more likely to diverge. An explanation for this influence is that the random seeds directly determine the values for weights when the neural network is initialized. Some of the seeds may lead to a more beneficial starting point for the following learning phase, compared to others that put the network very close to a local minimum.

In total, 20 random combinations are trained and evaluated at the end of each epoch using the DistilBERT for Sequence Classification architecture. The best model is picked taking into account the accuracy, precision, recall, F_1 and cross-entropy loss.

Figure 2 shows the combination of hyperparameters and achieved accuracy in each run. The models with the worst performance are the purple ones with an accuracy of slightly over 50%. The one with the second biggest learning rate has an accuracy of 86.5%, while all others have similar results: between 95% and 98%.

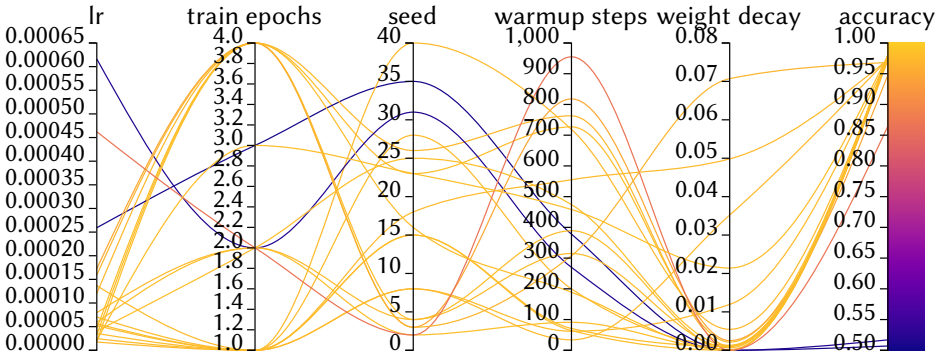


Figure 2: Summary of all performed runs

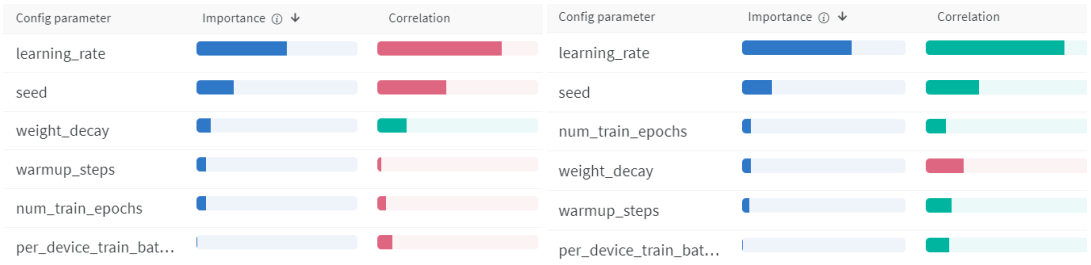
Considering the above metrics we choose our final model with the following rankings:

- Highest accuracy, F_1 score and recall
- Second smallest cross-entropy loss
- Sixth highest precision

The importance of each of the hyperparameters with respect to the accuracy can be seen in Figure 3, calculated by training a random forest classifier with the hyperparameters as inputs and the metric (in our case, accuracy and cross-entropy loss) as the target output and reporting the feature importance values for the random forest classifier. A negative correlation is colored in red, while a positive one is colored in green.

The learning rate has the biggest influence on the accuracy, and they have a negative correlation. The random seed has a decent level of importance, though not as high as the learning rate. The correlation between learning rate and loss is even more pronounced than that between learning rate and accuracy, and naturally, in the opposite direction. The only metric with an opposite correlation is the weight decay, but its importance with regard to both accuracy and loss is very low. All other metrics can be neglected.

An interesting observation is that this best model does not offer a big increase in accuracy compared to the out-of-the-box model, fine-tuned using the default parameters of the Train-



(a) Accuracy (b) Cross-entropy loss

Figure 3: Hyperparameter importance and correlation with respect to accuracy and cross-entropy loss

ingArguments class from Huggingface.[17] Despite that, it is clear that especially the learning rate plays a significant role in the model's performance.

6. Evaluation

6.1. Sentence Classification

We perform hyperparameter optimisation using DistilBERT (see Section 4). Below, we present the results from all models and explain why we chose DistilBERT for our further experiments.

6.1.1. Model Comparison

We compare four Transformer models, which are fine-tuned using the same parameters, using the HuggingFace TrainingArguments defaults. We use a batch size of 8 per device for training and 64 for evaluation, which is performed at the end of each epoch. The models are fine-tuned for two epochs. We take into account the metrics already mentioned in Section 5, i. e., precision, recall, F_1 score, accuracy and cross-entropy-loss.

Although RoBERTa has the highest performance in the GLUE benchmark, BERT achieves the highest accuracy and F_1 score and the lowest cross-entropy-loss in our case, see Table 1.

Table 1
Model comparison results

Model	Accuracy	CE loss	F_1	Precision	Recall
BERT	**98.11%**	**0.0952**	**0.9805**	0.9816	**0.9793**
DistilBERT	97.89%	0.09849	0.9781	0.9796	0.9773
SciBERT	97.64%	0.1197	0.9755	0.9799	0.9711
RoBERTa	97.61%	0.1006	0.975	**0.9818**	0.9684

Although the differences are minimal, SciBERT (our domain-specific model) and RoBERTa, which were possible favorites, deliver slightly worse results than DistilBERT and BERT.

[17] https://huggingface.co/transformers/main_classes/trainer.html

6.1.2. Cross-validation

We perform 10-fold cross-validation on the two best performing models (BERT, DistilBERT). Here, we only use our train dataset. In each of the ten iterations, 30% of the data is chosen randomly for testing only. Although BERT achieves higher accuracy in the simple comparison, the results during cross-validation are much closer, see Table 2.

Table 2
Cross-validation results

Model	Accuracy	CE loss	F_1	Precision	Recall
BERT	**97.7185%**	0.1216	**0.9769**	0.9762	**0.9777**
DistilBERT	97.692%	**0.0966**	0.9766	**0.9773**	0.976

Although BERT still has higher accuracy, this time the difference is much smaller (0.02%), and DistilBERT has a significantly lower cross-entropy loss and higher precision. Moreover, DistilBERT is fine-tuned at least twice as fast as BERT. For these reasons, we choose to further experiment with different hyperparameters for fine-tuning DistilBERT and pick the best configuration as the final model for the prototype UI. We manage to increase the accuracy of DistilBERT to 98.1%, the F_1 score to 98.03%, and minimize cross-entropy-loss to 0.08619. Such a high performance is possible, because the presented dataset simplifies the problem considerably: the suspicious sentences are mostly article headlines or individual statements from public figures. Consequently, they differ more from normal sentences and non-scandalous article headlines than more carefully hidden fake news would.

6.1.3. Qualitative Evaluation

As a first step of the qualitative evaluation, we compare our implementation to a similar tool – ClaimBuster, already mentioned in Section 2. In contrast to our approach, users have to set a threshold themselves between 0 and 1 to determine at which point claims are considered unreliable. We notice that many of the suspicious sentences are still not marked as such even with a rather low threshold of 0.3. Also, checking a claim is done via a separate search bar, which does not allow for fact-checking multiple claims at once. In comparison, our implementation not only performs these two steps automatically and for a whole text, but also with a higher accuracy. Moreover, it starts with binary classification, which is then extended to five classes after the claim validation step.

Since the evaluation dataset is not the perfect match for our use case, i. e., the fake news are often headlines, photo captions, etc., rather than sentences found in a full text, we perform a qualitative evaluation using a small number of articles. We use the "Global Social Responses to Covid-19 Web Archive"[18] and a couple of articles from reliable sources. The web archive includes social responses to the Covid-19 pandemic, as represented in local news, NGOs, art, blogs and social media. We focus mainly on articles from local news in English. The archive does not contain North American or West European sources, where the majority of fact-checkers are based. For this reason, although some sentences from local news are labeled as suspicious,

[18] https://archive-it.org/collections/14022

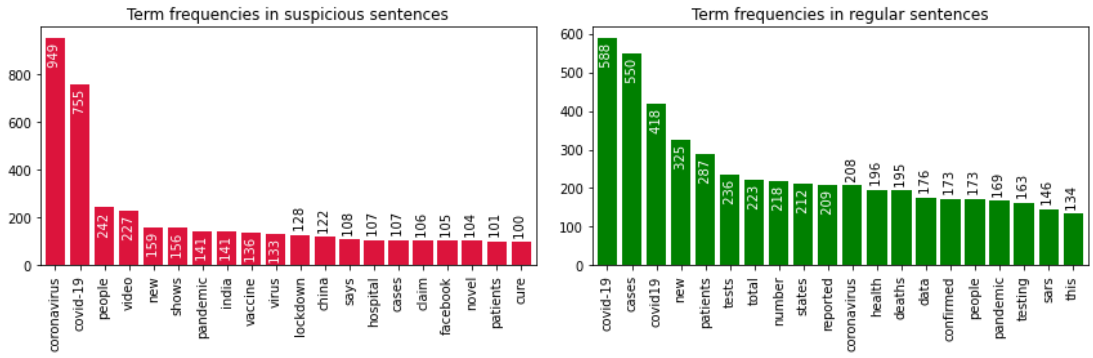


Figure 4: Top 20 most common terms within regular sentences and ones containing fake news

no relevant claims can be found for them. This uncovers the problem that suspicious statements from regional news are often not fact-checked at all, or at least not annotated using ClaimReview, and as a result, cannot be found using Google's tool.

Another observation is that hardly any sentences in articles from trusted sources, e. g., in the article "Coronavirus outbreak and kids" from Harvard Health Publishing[19], are labeled as suspicious. This is the expected behavior of the model.

With such complex model architectures, however, it is hard to determine which specifics of the data prove to have the biggest influence on the final predictions. We look at a couple of the misclassified examples (Table 3). A human annotator would possibly also classify the real example as dubious. However, the false negative is probably caused by the scientific terms used. This is the case for most of the wrongly classified examples, which stresses the point that a priority should be avoiding false negatives.

Table 3
Misclassified Examples

Sentence	True Label	Predicted Label
There is no one in New Zealand receiving hospital-level care for COVID-19.	regular	suspicious
Even discharged patients could be a long-term asymptomatic carriers.	suspicious	regular

In the hopes of discovering specific terms which can distinguish fake news from real ones, we calculate the word frequency in our validation dataset. The 20 most common words in each group of sentences can be seen in Figure 4.

Sentences from real news or scientific articles tend to prefer the term "COVID-19" instead of "coronavirus". Due to the nature of the suspicious sentences, typical words are "video", "lockdown", "facebook", etc. Also, we notice that verbs, such as "shows", "says" and "claim", are only present in the suspicious sentences, whereas the verbs among the top 20 of the regular

[19]https://www.health.harvard.edu/diseases-and-conditions/coronavirus-outbreak-and-kids

sentences are "states" and "reported". Altogether, the language in the non-suspicious sentences seems to be more factual and scientific, with more formal terms.

6.2. Claim Extraction

With regard to the evaluation of the claim extraction procedure we cannot measure the success of claim validation in a mathematical sense since the Google Claim Search API does not expect the claim to be structured data. While we can limit our results by choosing a language, a fact-checking website, etc., the claim itself is simply a text string. For this reason, we briefly measure the success of claim extraction/validation by the number of fetched relevant results. We search for claims relevant to each sentence in the validation dataset using Google's Fact Check Tools: once by using the full[20] sentence and once with the simplified sentence (without punctuation and stopwords). Surprisingly, there are slightly more cases in which no relevant results are found when removing the stopwords but choosing the more suitable approach depends on the type of sentence: in the case of, e. g., statements from or about known figures, full sentences deliver the same, if not better, results. However, when it comes to more general claims or more complex sentences, only keeping the keywords works better. Since the Google Claim Search API seems to work well with exact terms/structures, we decided not to perform additional processing, such as lemmatization or stemming.

Our approach, however, is not optimal for the following reason: we test using individual sentences, which are usually either a news title or a standalone statement, so they are not part of a text. This means that we do not have to deal with the problem of context, pronouns, etc. In these cases, removing them proves useful for finding more relevant results. Nevertheless, we are aware that future work should include a more sophisticated way of extracting claims.

7. Conclusion and Future Work

In terms of tackling the misinformation challenge, the approach we propose includes two separate steps: the identification of suspicious sentences (i. e., fact-check-worthiness estimation) followed by claim validation using trusted knowledge bases, both components can also be used separately. In comparison, existing approaches usually only implement one or the other, the middle ground between the two, i. e., claim extraction, is usually missing. Moreover, in this combined way, we can allow, at least to a certain degree, for some error in the first step, which can be corrected in the second, and we also introduce some nuance since each claim is categorised into one of five separate classes.

Our initial experiments, using datasets from different origin, proved not to adapt well in real-life cases. Using scientific articles as a source of non-suspicious sentences caused the problem that a large amount of non-scientific sentences were falsely labeled as suspicious and therefore unnecessarily fact-checked. One solution was bringing more variety to this part of the dataset by using real news. With this new dataset, curated using multiple sources of suspicious and non-suspicious sentences, much better results could be achieved.

[20] Here, we still remove punctuation since it often causes invalid queries.

As for claim extraction, a simple approach proved to be sufficient to find similar claims using the Google Fact Check Tools API. A problematic aspect was the deviation of the format of Google's claim search results from the standard ClaimReview markup. While a numerical rating is defined, only a textual one is present in what the tool provides. Moreover, fact-checkers are not required to stick to a certain format of their textual ratings, as even full sentences are allowed. As a result, we came up with a mapping from textual to numerical ratings.

All in all, our tool can help journalists and other people that rely on extensive fact-checking. It is fully automatic, easily accessible using an intuitive graphical interface and comes with a fast AI-based fake news detector. Thus, suspicious content can be detected in real time and validated against high-quality external databases. This procedure leads to a more productive fact-checking workflow, both in terms of speed and accuracy.

There are multiple directions in which our work can be taken further. They include more research on the performance of the models on full articles, Twitter threads, etc., extending the current datasets by adding data from full texts (instead of only titles and headlines), testing out different sequence classification models (including ensembles), and claim extraction approaches. Additionally, we plan to carry out a user study with journalists to assess if the developed prototype is helpful and if it makes them more efficient.

Acknowledgments

The research presented in this article was partially funded by the German Federal Ministry of Education and Research (BMBF) through the projects QURATOR (https://qurator.ai; grant no. 03WKDA1A) and PANQURA (http://qurator.ai/panqura; grant no. 03COV03E).

References

[1] C. Wardle, Fake news. It's complicated., https://firstdraftnews.com/fake-news-complicated/, 2017. First Draft News.

[2] W. Antoun, F. Baly, R. Achour, A. Hussein, H. Hajj, State of the art models for fake news detection tasks, in: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), IEEE, Doha,Qatar, 2020, pp. 519–524. doi:10.1109/ICIoT48696.2020.9089487.

[3] D. K. Vishwakarma, C. Jain, Recent state-of-the-art of fake news detection: A review, in: 2020 International Conference for Emerging Technology (INCET), IEEE, Belgaum,India, 2020, pp. 1–6. doi:10.1109/INCET49848.2020.9153985.

[4] P. Bourgonje, J. M. Schneider, G. Rehm, From Clickbait to Fake News Detection: An Approach based on Detecting the Stance of Headlines to Articles, in: O. Popescu, C. Strapparava (Eds.), Proceedings of the Second Workshop on Natural Language Processing meets Journalism – EMNLP 2017 Workshop (NLPMJ 2017), Copenhagen, Denmark, 2017, pp. 84–89. URL: http://www.aclweb.org/anthology/W/W17/W17-4215.pdf, 7 September.

[5] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, Science 359 (2018) 1146–1151. URL: https://science.sciencemag.org/content/359/6380/1146. doi:10.1126/science.aap9559.

[6] G. Bhatt, A. Sharma, S. Sharma, A. Nagpal, B. Raman, A. Mittal, Combining neural, statistical and external features for fake news stance identification, in: Companion Proceedings of the The Web Conference 2018, WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018, p. 1353–1357. URL: https://doi.org/10.1145/3184558.3191577. doi:10.1145/3184558.3191577.

[7] V.-H. Nguyen, K. Sugiyama, P. Nakov, M.-Y. Kan, Fang, Proceedings of the 29th ACM International Conference on Information and Knowledge Management (2020). URL: http://dx.doi.org/10.1145/3340531.3412046. doi:10.1145/3340531.3412046.

[8] K. Schulz, J. Rauenbusch, J. Fillies, L. Rutenburg, D. Karvelas, G. Rehm, User Experience Design for Automatic Credibility Assessment of News Content About COVID-19, in: HCI International 2022 – Late Breaking Papers, Springer, Virtual, 2022. Forthcoming.

[9] A. Srivastava, G. Rehm, J. M. Schneider, DFKI-DKT at SemEval-2017 Task 8: Rumour Detection and Classification Using Cascading Heuristics, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 486–490. URL: https://www.aclweb.org/anthology/S17-2085.pdf.

[10] G. Rehm, An Infrastructure for Empowering Internet Users to handle Fake News and other Online Media Phenomena, in: G. Rehm, T. Declerck (Eds.), Language Technologies for the Challenges of the Digital Age: 27th International Conference, GSCL 2017, Berlin, Germany, September 13-14, 2017, Proceedings, number 10713 in Lecture Notes in Artificial Intelligence (LNAI), Gesellschaft für Sprachtechnologie und Computerlinguistik e.V., Springer, Cham, Switzerland, 2018, pp. 216–231. URL: https://link.springer.com/content/pdf/10.1007%2F978-3-319-73706-5_19.pdf, 13/14 September 2017.

[11] G. Rehm, J. M. Schneider, P. Bourgonje, Automatic and Manual Web Annotations in an Infrastructure to handle Fake News and other Online Media Phenomena, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), Proceedings of the 11th Language Resources and Evaluation Conference (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018, pp. 2416–2422.

[12] B. Collins, D. T. Hoang, N. T. Nguyen, D. Hwang, Fake news types and detection models on social media a state-of-the-art survey, in: P. Sitek, M. Pietranik, M. Krótkiewicz, C. Srinilta (Eds.), Intelligent Information and Database Systems, Springer Singapore, Singapore, 2020, pp. 562–573.

[13] X. Li, Y. Xia, X. Long, Z. Li, S. Li, Exploring text-transformers in aaai 2021 shared task: Covid-19 fake news detection in english, in: T. Chakraborty, K. Shu, H. R. Bernard, H. Liu, M. S. Akhtar (Eds.), Combating Online Hostile Posts in Regional Languages during Emergency Situation, Springer International Publishing, Cham, 2021, pp. 106–115.

[14] S. Gundapu, R. Mamidi, Transformer based automatic covid-19 fake news detection system, 2021. arXiv:2101.00180.

[15] T. Mitra, E. Gilbert, Credbank: A large-scale social media corpus with associated credibility annotations, in: Ninth International AAAI Conference on Web and Social Media, The AAAI Press, Palo Alto, California, 2015, pp. 258–267.

[16] W. Y. Wang, "liar, liar pants on fire": A new benchmark dataset for fake news detection, in: Proceedings of the 55th Annual Meeting of the Association for Com-

putational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 422–426. URL: https://aclanthology.org/P17-2067. doi:10.18653/v1/P17-2067.

[17] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, H. Liu, Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media, Big Data 8 (2020) 171–188. URL: https://doi.org/10.1089/big.2020.0062. doi:10.1089/big.2020.0062. arXiv:https://doi.org/10.1089/big.2020.0062, pMID: 32491943.

[18] F. Arslan, N. Hassan, C. Li, M. Tremayne, A benchmark dataset of check-worthy factual claims, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 14, The AAAI Press, Online, 2020, pp. 821–829.

[19] A. Chernyavskiy, D. Ilvovsky, P. Nakov, Whatthewikifact: Fact-checking claims against wikipedia, in: CIKM '21: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Association for Computing Machinery, New York, NY, USA, 2021.

[20] A. Soleimani, C. Monz, M. Worring, Bert for evidence retrieval and claim verification, in: J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, F. Martins (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham, 2020, pp. 359–366.

[21] J. Thorne, A. Vlachos, C. Christodoulopoulos, A. Mittal, FEVER: a large-scale dataset for fact extraction and VERification, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 809–819. URL: https://aclanthology.org/N18-1074. doi:10.18653/v1/N18-1074.

[22] C. Wise, V. N. Ioannidis, M. R. Calvo, X. Song, G. Price, N. Kulkarni, R. Brand, P. Bhatia, G. Karypis, Covid-19 knowledge graph: Accelerating information retrieval and discovery for scientific literature, 2020. arXiv:2007.12731.

[23] D. Domingo-Fernández, S. Baksi, B. Schultz, Y. Gadiya, R. Karki, T. Raschka, C. Ebeling, M. Hofmann-Apitius, A. T. Kodamullil, COVID-19 Knowledge Graph: a computable, multi-modal, cause-and-effect knowledge model of COVID-19 pathophysiology, Bioinformatics 37 (2020) 1332–1334. URL: https://doi.org/10.1093/bioinformatics/btaa834. doi:10.1093/bioinformatics/btaa834.

[24] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, H. Hajishirzi, Fact or fiction: Verifying scientific claims, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 7534–7550. URL: https://aclanthology.org/2020.emnlp-main.609. doi:10.18653/v1/2020.emnlp-main.609.

[25] R. Vijjali, P. Potluri, S. Kumar, S. Teki, Two stage transformer model for covid-19 fake news detection and fact checking, in: G. Da San Martino, C. Brew, G. L. Ciampaglia, A. Feldman, C. Leberknight, P. Nakov (Eds.), Proceedings of the 3rd NLP4IF Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, International Committee on Computational Linguistics (ICCL), Barcelona, Spain (Online), 2020, pp. 1–10.

[26] 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.

[27] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A lite BERT for self-supervised learning of language representations, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, Addis Ababa, Ethiopia, 2020. URL: https://openreview.net/forum?id=H1eA7AEtvS.

[28] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, D. Zhou, MobileBERT: a compact task-agnostic BERT for resource-limited devices, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 2158–2170. URL: https://aclanthology.org/2020.acl-main.195. doi:10.18653/v1/2020.acl-main.195.

[29] A. Barrón-Cedeño, T. Elsayed, P. Nakov, G. Da San Martino, M. Hasanain, R. Suwaileh, F. Haouari, N. Babulkov, B. Hamdan, A. Nikolov, S. Shaar, Z. S. Ali, Overview of checkthat! 2020: Automatic identification and verification of claims in social media, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, H. Joho, C. Lioma, C. Eickhoff, A. Névéol, L. Cappellato, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction, Springer International Publishing, Cham, 2020, pp. 215–236.

[30] S. D. Das, A. Basak, S. Dutta, A heuristic-driven ensemble framework for covid-19 fake news detection, in: T. Chakraborty, K. Shu, H. R. Bernard, H. Liu, M. S. Akhtar (Eds.), Combating Online Hostile Posts in Regional Languages during Emergency Situation, Springer International Publishing, Cham, 2021, pp. 164–176.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in neural information processing systems, volume 30, Curran Associates, Inc., Red Hook, NY, USA, 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, ArXiv abs/1907.11692 (2019).

[33] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, in: Proceedings 2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS), IEEE Computer Society, Los Alamitos, CA, USA, 2019.

[34] I. Beltagy, K. Lo, A. Cohan, SciBERT: A pretrained language model for scientific text, 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), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3615–3620. URL: https://aclanthology.org/D19-1371. doi:10.18653/v1/D19-1371.

[35] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A Python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 101–108. URL: https://aclanthology.org/2020.acl-demos.14. doi:10.18653/v1/2020.acl-demos.14.

[36] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. M. Kinney, Y. Li, Z. Liu, W. Merrill, P. Mooney, D. A. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. Stilson, A. D. Wade, K. Wang, N. X. R. Wang, C. Wilhelm, B. Xie, D. M. Raymond, D. S. Weld, O. Etzioni, S. Kohlmeier, CORD-19: The COVID-19 open research dataset, in: Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Association for Computational Linguistics, Online, 2020. URL: https://aclanthology.org/2020.nlpcovid19-acl.1.

[37] G. K. Shahi, D. Nandini, FakeCovid – a multilingual cross-domain fact check news dataset for covid-19, in: Workshop Proceedings of the 14th International AAAI Conference on Web and Social Media, The AAAI Press, Atlanta, Georgia, USA, 2020. URL: http://workshop-proceedings.icwsm.org/pdf/2020_14.pdf.

[38] L. Cui, D. Lee, Coaid: Covid-19 healthcare misinformation dataset, 2020. arXiv:2006.00885.

[39] M. Claesen, B. D. Moor, Hyperparameter search in machine learning, ArXiv abs/1502.02127 (2015).

[40] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, I. Stoica, Tune: A research platform for distributed model selection and training, arXiv preprint arXiv:1807.05118 (2018).

[41] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, N. Smith, Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, 2020. arXiv:2002.06305.