OpenFact at CheckThat! 2023: Head-to-Head GPT vs. BERT - A Comparative Study of Transformers Language Models for the Detection of Check-worthy Claims

Notebook for the CheckThat! Lab at CLEF 2023

Marcin Sawiński[1,*,†], Krzysztof Węcel[1,†], Ewelina Księżniak[1], Milena Stróżyna[1,*], Włodzimierz Lewoniewski[1], Piotr Stolarski[1] and Witold Abramowicz[1]

[1]*Department of Information Systems, Poznań University of Economics and Business, Al. Niepodległości 10, 61-875 Poznań, Poland*

Abstract

This paper presents the research findings resulting from experiments conducted as part of the CheckThat! Lab Task 1B-English submission at CLEF 2023. The aim of the research was to evaluate the check-worthiness of short texts in English. Various methodologies were employed, including zero-shot, few-shot, and fine-tuning techniques, and different GPT and BERT models were assessed. Given the significant increase in the use of GPT models in recent times, we posed a research question to investigate whether GPT models exhibit notable superiority over BERT models in detecting check-worthy claims. Our findings indicate that fine-tuned BERT models can perform comparably to large language models such as GPT-3 in identifying check-worthy claims for this particular task.

Keywords

check-worthiness, fact-checking, fake news detection, language models, GPT, BERT, LLM

1. Introduction

In today's fast-paced and interconnected world, the need for fact-checking has become more critical than ever before. With the proliferation of social media platforms and the ease of sharing information, it has become increasingly challenging to discern between what is true and what is not. Rapid spreading of misinformation and disinformation for political, ideological,

CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

[*] Corresponding author.

[†] These authors contributed equally.

marcin.sawinski@ue.poznan.pl (M. Sawiński); krzysztof.wecel@ue.poznan.pl (K. Węcel); ewelina.ksiezniak@ue.poznan.pl (E. Księżniak); milena.strozyna@ue.poznan.pl (M. Stróżyna); wlodzimierz.lewoniewski@ue.poznan.pl (W. Lewoniewski); piotr.stolarski@ue.poznan.pl (P. Stolarski); witold.abramowicz@ue.poznan.pl (W. Abramowicz)

https://kie.ue.poznan.pl/en/ (M. Sawiński)

0000-0002-1226-4850 (M. Sawiński); 0000-0001-5641-3160 (K. Węcel); 0000-0003-1953-8014 (E. Księżniak); 0000-0001-7603-7369 (M. Stróżyna); 0000-0002-0163-5492 (W. Lewoniewski); 0000-0001-7076-2316 (P. Stolarski); 0000-0001-5464-9698 (W. Abramowicz)

or personal gains may lead to significant consequences on public opinion, decision-making processes, and even societal harmony [1]. This is why fact-checking plays a vital role, serving as a crucial tool to verify the accuracy and credibility of information. Initially, the primary focus of fact-checking was to confirm the accuracy of information presented in news articles prior to their publication. This responsibility lies at the heart of the journalistic profession. In present times, fact-checking refers also to the analyses of claims after a certain information is published and concerns particularly information that is shared on the Internet [2]. This is carried out by people (fact-checkers), not related to the author of information being verified, who critically examine and verify it and thus help to combat the spread of misinformation. Usually, the fact-checking process consists of several steps, starting with selecting a claim to check, through contextualizing and analyzing, consulting data and domain experts, writing up the results along with deciding on the rating, and finally disseminating the report [3]. The main challenge is that the majority of the fact-checker's job is still done manually. Therefore, there is a pressing need to develop various technologies that would facilitate, speed up, and improve fact-checking work and detection of fake news.

The first step of the fact-checking process primarily involves the identification of check-worthy claims. The aim is to identify, prioritize, filter, and select claims that are worth to fact-check, considering their factual coherence and potential impact. The process of selecting claims to fact-check entails identifying statements from diverse sources like posts, news articles, interviews, etc., assessing their check-worthiness (i.e., if they are factual claims that can be verified), and assessing their relevance and appeal to the target audience in terms of significance, usefulness, and engagement. In the research presented in this paper, our focus was specifically on the check-worthiness aspect. As outlined by [4], a claim may be deemed check-worthy if the information it carries is: 1) harmful – it attacks a person, organization, country, group, race, community, etc., or 2) urgent or breaking news – news-like statements about prominent people, organizations, countries and events, or 3) up-to-date – referring to recent official document with facts, definitions and figures. The automatic identification of such check-worthy claims is a challenging task and the main focus of this study.

The study presented in this paper is based on experiments performed as part of the CheckThat! Lab, Task 1B-English at CLEF 2023 [5]. The study focuses on a single task of assessing check-worthiness in unimodal (text-only) contents in English language. The task is defined as a binary classification problem, where the goal is to classify a given claim as check-worthy or not. The task is evaluated using the F1 score metric over positive class. The aim of this paper is to present various approaches that were applied in the task of assessing check-worthiness of unimodal content in English language and discuss the obtained results. It also shows the progress made beyond the state of the art.

The paper is organized as follows. Section 2 is an overview of state of the art in detecting check-worthy claims. Section 3 describes details of the conducted experiments, stating with dataset characteristic and how it was used for training. Methods based on GPT, BERT and boosting models follow. In Section 4 the results of experiments are discussed. The paper concludes with indications of directions for future work in Section 5.

2. Background

Transformer-based models, such as BERT and GPT-3, have been already used by teams that participated in CheckThat! Lab 2022 Task 1 that was held in the framework of CLEF 2022 [6]. This task concerned detection of relevant claims in tweets, taking into account various criteria, such as check-worthiness, verifiability, harmfulness, and attention-worthiness. The subtask, that concerned the check-worthiness of tweets, was the most popular one and covered several languages: Arabic, Bulgarian, Dutch, English, Spanish, and Turkish. However, English was the most popular target language. The datasets in all languages tackled tweets related to COVID-19 and politics. The evaluation metric for this task was the F1-measure with respect to the positive class. In total, there were 13 solutions submitted on the check-worthiness task for English language last year. The top-ranked system was built with RoBERTa large, after a data augmentation process based on back-translation [7], achieving F1-measure of 0.698. The tweets texts were translated to French, then back translated to English and combined to the training dataset. Moreover, all links from tweets were replaced with „@link". The second-best system was based on an ensemble approach that combined fine-tuned BERT and RoBERTa models (F1 of 0.667) [8]. In total, it was ensemble of ten models, pre-trained on tweets about COVID-19. Moreover, they applied various pre-processing techniques, like removing URLs, hashtags, numbers and other symbols. The third best solution, with F1 value of 0.626, used a fine-tuned GPT-3 model that is originally trained on English [9]. Other approaches, that were submitted or considered in internal experiments, were based either on a single transformer-based model, like BERT, DistilBERT, Electra, XML RoBERTa, mT5-XL, or XLNet, or ensemble of several models, e.g., various versions of BERT and RoBERTa models. There were also tested solutions with classifiers, like SVM and Random Forest. Moreover, most of the teams applied various additional techniques, starting from data augmentation to increase the size of training dataset (e.g., machine translating labeled datasets in other languages to the corresponding language, or back-translation), feature extraction for tweets, which were further used in addition to the textual data, using ELMo embeddings combined with linguistic features (LIWC), or including additional unlabeled training data. There were also some experiments with quantum natural language processing (QNLP), however the technique posed some problems, as reported by [10].

It is worth-mentioning that some solutions covered multiple languages, by application of different strategies, such as MT-based data augmentation (application of translation and back-translation to increase the training dataset in different languages) [11], mT5 multilingual transformer (a single model that might be applied to multiple languages) [12], or zero-shot strategy (a fine-tuned GPT-3 model fed with only instances in English and applied to other languages during testing) [9].

3. Experiments

The main objective of the experiments was to verify the hypothesis that the large GPT models are able to significantly outperform BERT models in detecting check-worthy claims. In order to test the hypothesis, multiple experiments were carried-out using various GPT and BERT models, as described in the following sections.

3.1. Dataset

The dataset offered for training consisted of 23,533 statements extracted from U.S. general election presidential debates, annotated by human coders and originally published in January 2015, known as ClaimBuster dataset [13]. This is not completely inline with the subtask description that texts in dataset are multigenre.

The dataset was split into *train*, *dev* and *dev_test* with 16,876, 5,625 and 1,032 examples in each split respectively. The comparison with the original ClaimBuster dataset revealed that *train* and *dev* splits were generated from examples with *crowd-sourced* labels. The *dev_test* split was identical to the ClaimBuster dataset, called *ground-truth*.

The *ground-truth* dataset was labeled by 3 experts and was used to screen spammers and low-quality participants of the *crowd-sourced* part of the dataset. The difference between *ground-truth* and *crowd-sourced* surfaced during evaluation of the results. The models trained on *train* dataset achieved, on average, F1 score 0.1 higher when tested on *dev_test* (i.e., *ground-truth*) then on *dev* (i.e., *crowd-sourced*). The difference could be attributed to the composition of split (e.g., less borderline examples in *dev_test*) or the quality of labels (e.g., a higher consistency in *dev_test*), with the latter being more probable as it correlates with the dataset creation process (i.e., experts vs. crowd-sourced labels).

This observation led to the conclusion that reshuffling the splits and filtering of the dataset could be a way to avoid overfitting to the *crowd-sourced* labels and to improve the model predictions.

3.2. Methods based on GPT models

Transformers have revolutionized NLP field and became a dominant architecture for state-of-the-art solutions, including the top-ranked solutions submitted to CheckThat! Lab in recent years. In 2022 most teams used BERT models [6] and only one team [14] used GPT models in their solution and won 1st place in the subtask 1B - 'Verifiable Factual Claims Detection' and 3rd place in the subtask 1A - 'Check-Worthiness Estimation' in English language. Given the spectacular development of GPT models over the recent year, we decided to verify the potential of large GPT models in detecting check-worthy claims. The study explored three main approaches to using GPT models: zero-shot learning, few-shot learning, and fine-tuning.

GPT models can be trained in two distinct ways: for text completion and for following instructions while maintaining conversation (chat). At the time of conducting the experiments, many pre-trained GPT models were available from many authors, e.g., OpenAI (GPT-3[1], GPT-3.5[2], GPT-4[3]), Anthropic (Claude[4]), Stanford(Alpaca)[15], Meta(LLaMA)[16]. The models varied in size, number of parameters, training data, and training objectives.

OpenAI models were chosen for experiments for two reasons: they showed the advantage in multiple performance tests and they were cost-effective in terms of fine-tuning and inference (e.g., the cost of fine-tuning a GPT-3 based curie model with 7000 examples via API was around

[1] https://platform.openai.com/docs/models/gpt-3

[2] https://platform.openai.com/docs/models/gpt-3-5

[3] https://platform.openai.com/docs/models/gpt-4

[4] https://www.anthropic.com/index/introducing-claude

2 USD, while a creation of a dedicated Virtual Machine with NVidia GPU A100 80GB to host a model similar in size would cost more by a few orders of magnitude). However, we should keep in mind that the cloud-based models available for prompting and fine-tuning via API can change over time and impact reproducibility of experiments.

The GPT language models created by OpenAI in 2018 use transformers architecture together with generative pre-training on a large corpus of unlabelled text, followed by discriminative fine-tuning on specific tasks. The subsequent updates to the GPT model formed a series of "GPT-n" models. The original GPT-1 model, released in 2018, had 117 million parameters and was trained on BooksCorpus dataset (7 000 unique unpublished books from a variety of genres, 4.5 GB) [17].

The GPT-2 model, released in 2019, introduced modified initialization, pre-normalization, and reversible tokenization. It featured 1.5 billion parameters and was trained on WebText dataset (40 GB) [18].

The GPT-3 model, released in 2020, introduced alternating dense and locally banded sparse attention patterns in the layers of the transformer [19]. It featured 175 billion parameters and was trained on a filtered and deduplicated Common Crawl dataset (570GB) and other high-quality reference corpora (WebText2, Books and Wikipedia). Additionally, a set of 8 differently sized models was created to test dependence of model performance on its size. Four models were released for general use and named: ada, babbage, curie, and davinci with 350 million, 3, 13 and 175 billion parameters, respectively. At the time of conducting the experiments presented in the paper, the four GPT-3 models were the most advanced language models from OpenAI that were publicly available for fine-tuning. The datasets used to train these models might have changed since the publication of the original paper. At the moment of writing Microsoft reports, the curie model was trained using 800GB of text data and the davinci model was trained using 45TB of text data[5].

Further on, fine-tuning using a combination of supervised training and reinforcement learning from human feedback allowed for the creation of instruction-following models (InstructGPT) that could be further fine-tuned for conversational interaction (ChatGPT). The GPT-3.5 model, released in 2022, is optimized for dialogue and forms the basis for ChatGPT. The size and performance of the model is comparable to Instruct Davinci (i.e., the biggest GPT-3 model fine-tuned for instruction-following; however, the technical details of the model were not disclosed by OpenAI).

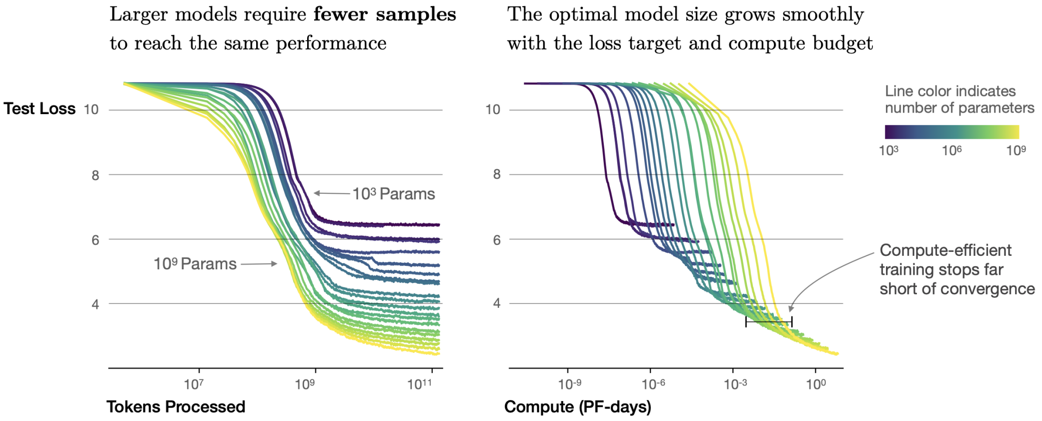
The GPT-4 model, released in 2023, is also fine-tuned for instruction-following and for dialogue. It brought a significant improvement over GPT-3.5 in numerous benchmarks, but the technical details of the model were not disclosed by OpenAI [20].

In our experiments, the GPT-3 model was used for fine-tuning and the GPT-3.5 and GPT-4 models were used for zero-shot and few-shot learning. The text completion approach is best used with GPT-3 models for fine-tuning, while the chat approach can leverage the GPT-3.5 and GPT-4 models for using zero-shot learning and few-shot learning. The GPT models do not require *text preprocessing* and were used with raw text from the dataset.

Data augmentation techniques, that often improve the performance of the model by enhancing

[5] https://learn.microsoft.com/en-us/semantic-kernel/prompt-engineering/llm-models

the dataset, were not applied. According to the OpenAI fine-tuning guide[6], each doubling of the dataset size leads to a linear increase in model quality. However, in our experiments, we have not tested this assumption and decided to go in the opposite direction. Our experiments were inspired by intuition derived from Kaplan et al. [21] that show: bigger models do not scale linearly with the size of the training data. The conclusion that a large enough model can be trained on a small dataset and still achieve good results led to an idea to explore the potential of limiting the size of the dataset in order to improve its quality. The reasoning is illustrated in Figure 1.



Larger models require **fewer samples** to reach the same performance

The optimal model size grows smoothly with the loss target and compute budget

Figure 1: A series of language model training runs, with models ranging in size from 10^3 to 10^9 parameters (excluding embeddings). Source: Scaling Laws for Neural Language Models [21]

3.2.1. Zero-shot learning using GPT-4

Zero-shot learning is a technique used in machine learning, particularly in natural language processing (NLP), where models are able to perform tasks without having prior examples of the specific task in their training data. GPT-4 is designed to have a good understanding of language and context, enabling it to perform various tasks with no task-specific fine-tuning.

To use this approach, the input prompt is designed to give clear and direct instructions related to the task [22]. These instructions help the GPT model understand the desired output format and perform the task using its pre-trained knowledge. The more specific and contextually relevant the instructions are, the better the model performs.

The input prompt for zero-shot learning with GPT-4 is shown in Listing 1. The variable `{claims}` in the prompt template is substituted during the run-time with the numbered list of claims to be classified (see Listing 2). The output of the model is a numbered list of check-worthiness ratings for each claim in the input list (see Listing 3).

Listing 1: Zero-shot learning: the prompt template for GPT-4

```
You are a factchecker assistant with task to identify
   sentences that are check-worthy. Sentence is check-
```

[6] https://platform.openai.com/docs/guides/fine-tuning

```
    worthy only if it contains a verifiable factual claim
    and that claim can be harmful.

Classify the check-worthiness of these sentences outputting
    only yes or no:
{claims}
Check-worthiness ratings:
```

Listing 2: Zero-shot learning: an example of input for the `{claims}` variable

```
1. He’s been a professor for a long time at a great school.
2. There’s no way they would give it up.
3. They’re able to charge women more for the same exact
   procedure a man gets.
4. As far as a say is concerned, the people already had
   their say.
5. I am the Democratic Party right now.
```

Listing 3: Zero-shot learning: an example of output for the prompt

```
1. Yes
2. No
3. Yes
4. No
5. Yes
```

The prompt template was designed to leverage the knowledge accumulated in the model weights during pre-training. To avoid the ambiguity of the term *check-worthiness*, the prompt instructs the model to classify the sentences as check-worthy or not based on specific criteria. The check-worthiness definition is briefly explained (the claim must be factual, verifiable, and potentially harmful). Please note that the terms factual, verifiable, and harmful are not further explained to the model and the prompt relies on the pre-trained representations of these concepts. To limit the token count of the prompt, the model is instructed to output only values *Yes* or *No* and takes multiple sentences as input. We have observed that changing of the wording of the prompt has a visible impact on individual responses; however, the overall classification accuracy remains at a similar level. The same applies to minor spelling or grammatical errors.

3.2.2. Few-shot learning using GPT-4

Few-shot learning refers to the ability of a machine learning model to learn and adapt to new tasks or predict outputs based on a small amount of training data. While traditional deep learning techniques generally rely on extensive data for successful task performance, the few-shot learning method in the GPT model utilizes the pre-trained weights to reduce the number of training examples required. It is important to note that the model is not trained (fine-tuned) on the new task, i.e., the model weights are not updated, but the expected behavior is formed only during the inference time [19].

Few-shot learning experiment used three prompt templates:

- A *system prompt* that guides the model regarding the expected response by giving a context and an instruction to initiate the model's output. The system prompt is the same for all claims in the input list (see Listing 4).
- A list of *assistant prompts* that are used to convey the examples of the expected output. The assistant prompts can be different for each claim in the input list to reflect a specific context (see Listing 5).
- A *user prompt* that is used to initiate the model's output. It contains the claim to be classified and the expected output format (see Listing 6).

In order to find the best examples to be included in the few-shot learning experiment, the semantic search approach was applied. Firstly, all sentences from the train dataset were converted into sentence embeddings (768 dimensional dense vectors) using all-mpnet-base-v2 model from the HuggingFace Transformers library[7]. Secondly, the embeddings were used to find the most similar examples to the claim to be classified. Eight most similar examples (the top four with the label *Yes* and the top four with the label *No*) were then used as the assistant prompts. The similarity was calculated using the cosine similarity measure.

Listing 4: Few-shot learning: the system prompt template

```
You are a fact-checker assistant with task to identify
   check-worthy sentences.
You need to evaluate 2 conditions:
condition 1 - sentence contains a verifiable factual claim,
condition 2 - the claim could be potentially harmfull.
A sentence is check-worthy if both conditions are met - in
   this case expected output is 'yes'.
Otherwise output 'no'.
```

Listing 5: Few-shot learning: an example of the assistant prompt

```
User: I want it to be clearer.
Assistant: No

User: Iran went from zero centrifuges to develop nuclear
   weapons to 4,000.
Assistant: Yes

User: Uh - Also, so that we can get the uh - opportunity
   for the questioners to question me, it will be before
   the next television debate.
Assistant: No

User: Marijuana use is up 141percent.
```

[7] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

```
    Assistant: Yes

    User: Well, Alan (ph), thank you very much for the question
       .
    Assistant: No

    User: When they tried to reduce taxes, he voted against
       that 127 times.
    Assistant: Yes
```

Listing 6: Few-shot learning: an example of the user prompt

```
    User: When they tried to reduce taxes, he voted against
       that 127 times.
    Assistant:
```

3.2.3. Few-shot learning with Chain-of-Thought using GPT-4

The popular method to improve the quality of responses generated by sufficiently large language models is to use the Chain-of-Thought approach [23]. This approach builds on the few-shot learning method and extends it by decomposing complex, multi-step problems into intermediate steps, which can form a predefined protocol for solving a task. The Chain-of-Thought experiment was conducted using only four examples (two with the label *Yes* and two with the label *No*). The examples were selected manually and used as the assistant prompts (see Listing 7).

Listing 7: Few-shot learning: an example of the user prompt

```
Question: Classify if the  sentence is check-worthy: They have
   a VAT tax.

Are follow up questions needed here: Yes.
Follow up: Does the sentence contain a verifiable factual claim
   ?
Intermediate answer: Yes
Follow up: Could the claim be considered a statement of opinion
   ?
Intermediate answer: No
Follow up: What is the broad topic category of the claim?
Intermediate answer: Economy.
Follow up: Is the topic category sensitive?
Intermediate answer:  Yes
Follow up: Can the claim be harmful if false?
Intermediate answer: Yes.
So the final answer is: Yes.
```

```
Question: Classify if the sentence is check-worthy: That's one
   way one could do it.
```

3.2.4. Fine-tuning GPT-3

As mentioned earlier, at the time of conducting the experiments, the fine-tuning was available for the following base models from OpenAI[8]: davinci, curie, babbage, and ada based on the GPT-3 architecture. These models do not have any instruction-following training. Instead, in the fine-tuning process, the model is trained to generate text that is similar to the provided examples. The dataset received from CheckThat! 2023 organizers was split into *train*, *dev* and *dev_test* with the major difference observed in *dev_test* (see Section 3.1).

Since the larger models tend to learn from fewer examples [21], the dataset was truncated to exclude labels of the lowest quality. The authors of the ClaimBuster dataset introduced screening criteria to exclude low quality labels and published the same dataset filtered using stricter criteria with respect to the labels assigned (the file called 2xNCS.json[9]). This reduced dataset was used for fine-tuning GPT-3 models, considering the authors' experience that they produce better models. In order to capture the effect of the train data quality on the fine-tuned model predictions, two distinct datasets were prepared for fine-tuning with a total of 8706 examples each. The first model (called *curated*) was fine-tuned on the dataset with the highest quality labels (the examples also mentioned in ClaimBuster 2xNCS.json file) and the second model (called *raw*) was fine-tuned on the dataset with randomly selected examples. The dataset split was 1000 examples for validation and 7706 for training. After removing duplicated entries, the final train dataset contained 7694 and 7685 examples respectively. It is important to note that no new examples or features were added to the dataset and the whole training and validation process was executed using the dataset provided by the CheckThat! 2023 organizers. The only information derived from ClaimBuster was the list of examples ids with expected higher quality of labels that were used to filter the dataset further utilized for fine-tuning *curated* version on the model.

The fine-tuning was performed using davinci and curie models from OpenAI. The same hyperparameter values were used for all fine-tuning experiments (see Table 1). The format of the data prepared for fine-tuning is shown in Listing 8. Data used for fine-tuning contained only a binary label (*Yes* or *No*) without any indication of the purpose of the classification. This is important because the fine-tuning process is not aware of the task that the model is supposed to solve, yet it is still able to learn to classify the sentences correctly.

Listing 8: Fine-tuning data format example

```
{"prompt":"We might have to do it slowly. ->","completion
   ":" no\n"}
{"prompt":"It was $200 billion deficit instead. ->","
   completion":" yes\n"}
```

[8] https://platform.openai.com/docs/models/gpt-3

[9] https://zenodo.org/record/3836810

Table 1
Hyperparameters used for fine-tuning GPT-3 models

Hyperparameter	Value
Batch size	8
Learning rate multiplier	0.1
Epochs	4
Prompt loss weight	0.01
Compute classification metrics	True

3.3. Methods based on BERT models

In this section, we describe the methodology of fine-tuning various BERT-based models for the task of classifying claims as check-worthy or not.

3.3.1. Basic models

Datasets: We used datasets as specified in the CheckThat! Lab, Task 1B-English, without reshuffling. `train.tsv` with 16,876 rows was used for training, `dev.tsv` with 5625 rows was used as evaluation during training, and `dev_test.tsv` with 1032 rows to compare the performance of various trained models.
Hardware: we fine-tuned our models on a local machine with four NVIDIA GeForce RTX 2080 Ti GPU cards, each with 11 GB of memory. The available machine limited the number of potential models that might be used for fine-tuning. Our experience shows that larger models usually caused 'out of memory' error (OOM).

The following models were subject to the fine-tuning: • DistilBERT [24], • DeBERTa [25], • RoBERTa [26], • XLM-RoBERTa [27], • ALBERT [28], • RemBERT [29], • CamemBERT [30], • ELECTRA [31], • GPT neo 125M [32], • YOSO [33]. Although the CamemBERT model is for French, we wanted to check, how it will behave during the fine-tuning. YOSO was the most exotic one among the tested models, and an efficient sampling scheme for short claims did not yield any satisfactory results (during the evaluation, some of the answers even appeared to have swapped "Yes" and "No" responses).

The default optimization method for training was AdamW (the Pythorch version `adamw_pytorch`). When the model was larger and caused OOM, we first reduced the batch size. Typically, we decreased from 16 to 8, or even to 4 in extreme cases. In the case of small batches, we also turned on a variant based on the gradient accumulation with 8 or 16 steps. It increased the training time without a significant impact on the accuracy. Concerning the float precision, we tested two settings for the models: FP16 and FP32. However, this did not result in an observable reduction in memory usage. Training could be faster, though. When the above approaches to reduce memory usage were not sufficient, we also switched the optimizer to Adafactor, but this only applied to the RemBERT model. Adafactor reduces memory usage while retaining the empirical benefits of adaptivity. Using Adafactor allowed us to obtain results, although the training time was much longer, reaching 3 hours, whereas typical fine-tuning with AdamW was 20-30 minutes.

Various learning rates were also tested. The default learning rate (`lr`) was 2e-5, and alternatives values of 1e-5 and 3e-5 were also explored. The typical fine-tuning duration was scheduled for 5 epochs and took 20-30 minutes. If it was observed that the evaluation loss was still decreasing, the training was extended to 10 epochs while monitoring for signs of overfitting. The following models were trained for 10 epochs: RoBERTa, DistilBERT, YOSO, GPT-neo, and RemBERT with Adafactor. Aditionally, more sophisticated methods for the fine-tuning were applied, such as layer-wise learning rate decay [34]. This method involves using different learning rates for different layers. The rationale behind this approach is that the lower layers (closer to the input) capture general language information, while the upper layers (closer to the output) encode task-specific information, such as classification. In general, the upper layers are trained with a higher learning rate, which gradually decreases when moving to the lower layers. The decay rate assumed in our method is 0.9. One of the RoBERTa models was trained using this custom optimizer.

3.3.2. F1 as training objective

Optimization of a classification model typically focuses on accuracy. However, tn the defined Task 1B, the criteria for model selection was a higher F1 for the positive class. Models were fine-tuned with this objective in mind, but it did not always result in the best results when evaluated on the test dataset.

In Figure 2, two confusion matrices are shown: one is of the model trained to maximize F1 macro average (on the left), and the other is of the model trained specifically to maximize F1 of the positive class 'Yes' (on the right). The test dataset used for this and the following evaluations we was `dev_test.tsv`, which contains 1032 rows.

The DeBERTa model depicted on the left of Figure 2 was also the best among locally trained BERT-like models in terms of the final evaluation. It was only 0.004 worse in terms of F1 measure compared to the overall winning model.

	precision	recall	F1-score	support
No	0.98	0.98	0.98	794
Yes	0.95	0.94	0.95	238
accuracy			0.97	1032
macro avg	0.97	0.96	0.96	1032
weigh. avg	0.97	0.97	0.97	1032

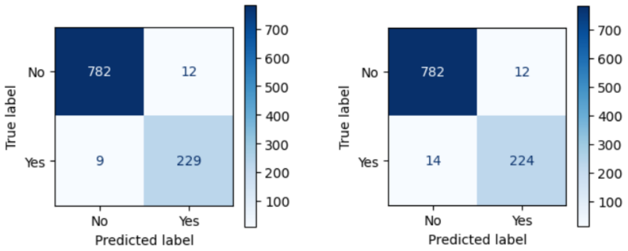


Figure 2: Metrics and confusion matrices for DeBERTa based models trained with various objectives. Left: F1 macro avg. Right: F1 positive optimized.

3.3.3. Layer-wise learning rate decay

Here, we present the results of applying the layer-wise learning rate decay method. In Figure 3, two confusion matrices are shown. The left matrix presents the RoBERTa model trained with a fixed learning rate, i.e., it was the same for each layer. On the right, the matrix presents the RoBERTa model trained with varying learning rate, where the learning rate decreases as we move closer to the bottom layers. As expected, the more sophisticated optimizer provided a better model, at least on the dev_test datasets. The number of false positives was reduced by 3, and the number of false negatives was reduced by 6. The final results on the test dataset were comparable – the difference was only 0.002 in the F1 score.

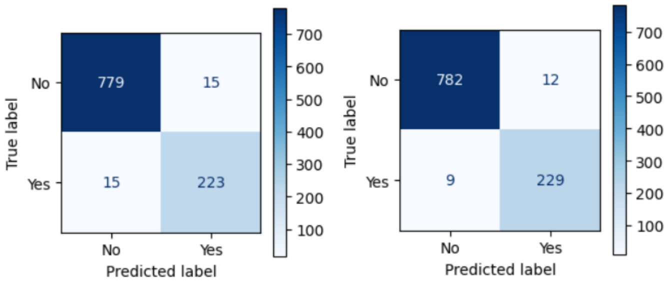


Figure 3: Confusion matrices for the RoBERTa based models trained with various learning rates. Left: fixed learning rate. Right: layer-wise learning rate decay.

3.3.4. The best model according to dev_test dataset

The most promising results were provided by the ELECTRA model. It had the highest value of F1 score for the positive class. The evaluation metrics and confusion matrix on dev_test dataset are presented in Figure 4.

	precision	recall	F1-score	support
No	0.99	0.99	0.99	794
Yes	0.96	0.97	0.97	238
accuracy			0.98	1032
macro avg	0.98	0.98	0.98	1032
weighted avg	0.98	0.98	0.98	1032

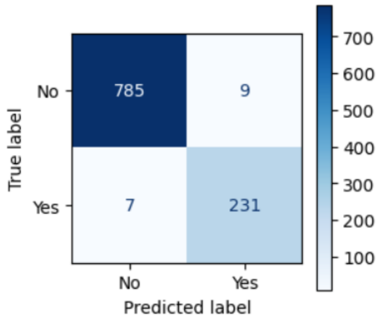


Figure 4: Metrics and confusion matrix of the ELECTRA model

However, in the final evaluation, the model did not perform so well. It was worse by 0.047 compared to the best model with regards to F1 score of the positive class.

3.4. LightGBM ensemble model

Analysis of submissions to previous editions of the CheckThat! Lab [4, 35, 6] and similar publications suggested exploring *ensemble methods* to improve prediction accuracy. As a result, we decided to investigate the suitability of the LightGBM model, a gradient boosting framework that utilizes tree-based learning algorithms. The approach for this model was to combine the outputs from the previous fine-tuned models, along with predictions from other available models (data enrichment). This data included both the predicted labels and their corresponding probabilities.

Our LightGBM mode was trained on the following features:

- predictions (0 or 1) and probabilities (between 0.0 and 1.0) from the following fine-tuned models: *roberta_pred, roberta_probY, roberta_probN, roberta2_pred, roberta2_probY, roberta2_probN, xlm-roberta_pred, xlm-roberta_probY, xlm-roberta_probN, deberta_pred, deberta_probY, deberta_probN, deberta2_pred, deberta2_probY, deberta2_probN, distilbert_pred, distilbert_probY, distilbert_probN, albert_pred, albert_probY, albert_probN, electra_pred, electra_probY, electra_probN, yoso_pred, yoso_probY, yoso_probN, gptneo_pred, gptneo_probY, gptneo_probN, BERTemo_pred, albert2_pred, albert2_probY, albert2_probN*;
- emotion probability calculated with BERTemo[10]: *BERTemo_sadness, BERTemo_joy, BERTemo_love, BERTemo_anger, BERTemo_fear, BERTemo_surprise*;
- sentiment probability calculated with ReBERTa for sentiment analysis[11]: *RoBERTasent_pred, RoBERTasent_negative, RoBERTasent_neutral, RoBERTasent_positive*;
- logits returned by ELECTRA discriminator[12]: *ELECTRA_logit_first, ELECTRA_logit_last, ELECTRA_logit_min, ELECTRA_logit_avg, ELECTRA_logit_max, ELECTRA_num_odd, ELECTRA_pcnt_odd.*

The last point requires special attention. The ELECTRA model is trained with the objective of detecting replaced tokens. By utilizing the discriminator, it is possible to calculate the probability of each token (typically a word, depending on tokenization) in the sentence. If an unexpected token occurs, it may serve as a signal to verify the sentence. The assumption is that anomalous sentences are more lie=kely to be check-worthy. From the ELECTRA discriminator, we obtain logits for each token in the input sentence. We then create the following variables: the logit of the first token, the logit of the last token, the minimum logit value, the mean logit value, the maximum logit value, the number of odd tokens (when logit is bigger than zero), and the percentage of odd tokens.

We have explored various settings for hyperparameters in numerous attempts. However, despite our efforts, the F1 score never exceeded 0.79 on the evaluation set (`dev_test.tsv`, N=1032). Many training sessions exhibited similar patterns to the one depicted in Figure 5. The occurrence of overfitting varied depending on the learning rate, with some cases experiencing it sooner while others later in the training process.

Training of the best model was conducted with the following hyperparameters: objective: binary, boosting: dart, learning_rate: 0.05, bagging_fraction: 1, max_depth': -1, num_leaves: 255, min_gain_to_split: 0, min_data_in_leaf: 20, min_sum_hessian_in_leaf: 0.001, max_bin: 255, num_leaves: 255.

[10] https://huggingface.co/bhadresh-savani/bert-base-uncased-emotion

[11] https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest

[12] https://huggingface.co/google/electra-large-discriminator

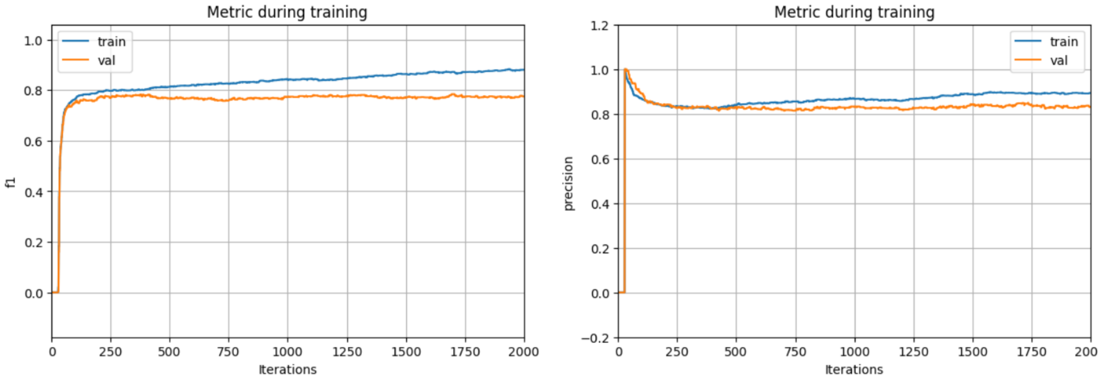


Figure 5: Training of Light GBM models: all 4 metrics

Concerning the importance of variables, the following were the most important: deberta_probY=5226 and xlm-roberta_probY=4676. So, the best models also contributed the most to the ensemble model. Our additional models, beyond the fine-tuned ones, also contributed to the overall result. For example, BERTemo_love=3971 was on the fourth place; the ninth place was taken by ELECTRA_logit_last=3363, and RoBERTasent_neutral=3276 was on the tenth place.

4. Summary of experiments results

4.1. Metrics calculated on test set

The highest F1 score of 0.898 was obtained by the GPT-3 curie model, which was fine-tuned with approximately 7690 examples selected based on stricter label quality criteria (GPT-3 curie fine-tuned curated). The model demonstrated top performance during internal validation and was subsequently submitted for the CheckThat! 2023 evaluation, where it achieved the highest score.

Interestingly, when the same model (GPT-3 curie) was fine-tuned with identical hyperparameters and the same number of examples (approximately 7690) randomly sampled from the entire dataset, it obtained F1 score of only 0.826 (GPT-3 curie fine-tuned random). The difference between these two models is significant (0.072) and shows that the quality of fine-tuning data is crucial for the final performance of the model.

Surprisingly, the larger GPT-3 davinci model, trained with the same setup and dataset as the winning method, performed marginally worse across all metrics, obtaining the F1 score of 0.876 (GPT-3 davinci fine-tuned curated), despite having 13 times more parameters.

The second best model was DeBERTa v3 with F1 score of 0.894 (DeBERTa v3 base fine-tuned). With only 86M backbone parameters, 98M parameters in the Embedding layer, and trained on 160GB data, this model displayed unparalleled parameter-efficiency, when compared to much larger curie and davinci models. Despite having the same accuracy as the winning model, the F1 score was only 0.004 lower due to a slightly less favorable precision-recall trade-off.

Other BERT models showed lower performance, but still surpassed the scores of zero-shot learning and few-shot learning approaches achieved by GPT-4.

The complete results are presented in Table 2, and further can be found in the confusion matrices of the test results generated for each model (Figure 6).

Table 2
The results obtained by the GPT and BERT models

Model	F1	precision	recall	accuracy
GPT-3 curie fine-tuned curated	0.898	0.948	0.852	0.934
DeBERTa v3 base fine-tuned	0.894	0.978	0.824	0.934
GPT-3 davinci fine-tuned curated	0.876	0.946	0.815	0.921
RoBERTa base fine-tuned	0.862	0.966	0.778	0.915
RoBERTa base fine-tuned with custom optimizer layer-wise learning rate decay	0.860	0.976	0.769	0.915
LightGBM ensemble of all BERT-based models and additional embeddings	0.854	0.976	0.759	0.912
ELECTRA fine-tuned	0.851	0.954	0.769	0.909
AlBERT large v2 fine-tuned	0.848	0.976	0.750	0.909
DistilBERT base uncased fine-tuned	0.827	0.952	0.731	0.896
GPT-3 curie fine-tuned random	0.826	1.000	0.704	0.899
GPT neo 125M fine-tuned	0.800	0.961	0.685	0.884
GPT-4 few-shot learning	0.788	0.867	0.722	0.868
GPT-4 zero-shot learning	0.778	0.710	0.861	0.833
GPT-4 Chain-of-Thought	0.722	0.574	0.972	0.745

4.2. Analysis of the results

It is challenging to determine a clear winner between the BERT and GPT models. The leading models, namely GPT-3 curie and DeBERTa v3, performed comparably throughout the competition, consistently yielding similar results during validation and testing. The GPT-3 curie model was selected for the final submission due to its slightly better performance on the validation dataset, although the difference was not significant. However, in real-world scenarios, the DeBERTa v3 model may be preferred due to its smaller size and faster inference time. Additionally, DeBERTa v3 exhibited an advantage over the CheckThat! Lab 2022 Task 1 winning model, RoBERTa, proving the ongoing advancements in BERT architectures that are worth further exploration.

On the other hand, the GPT models demonstrated exceptional capabilities in leveraging limited data through few-shot and zero-shot learning. Although the achieved performance in this competition, with F1 scores of 0.788 and 0.778 respectively, may not appear remarkable, these results could be considered satisfactory when applied to real-world scenarios involving unknown data and the absence of an extensive training set.

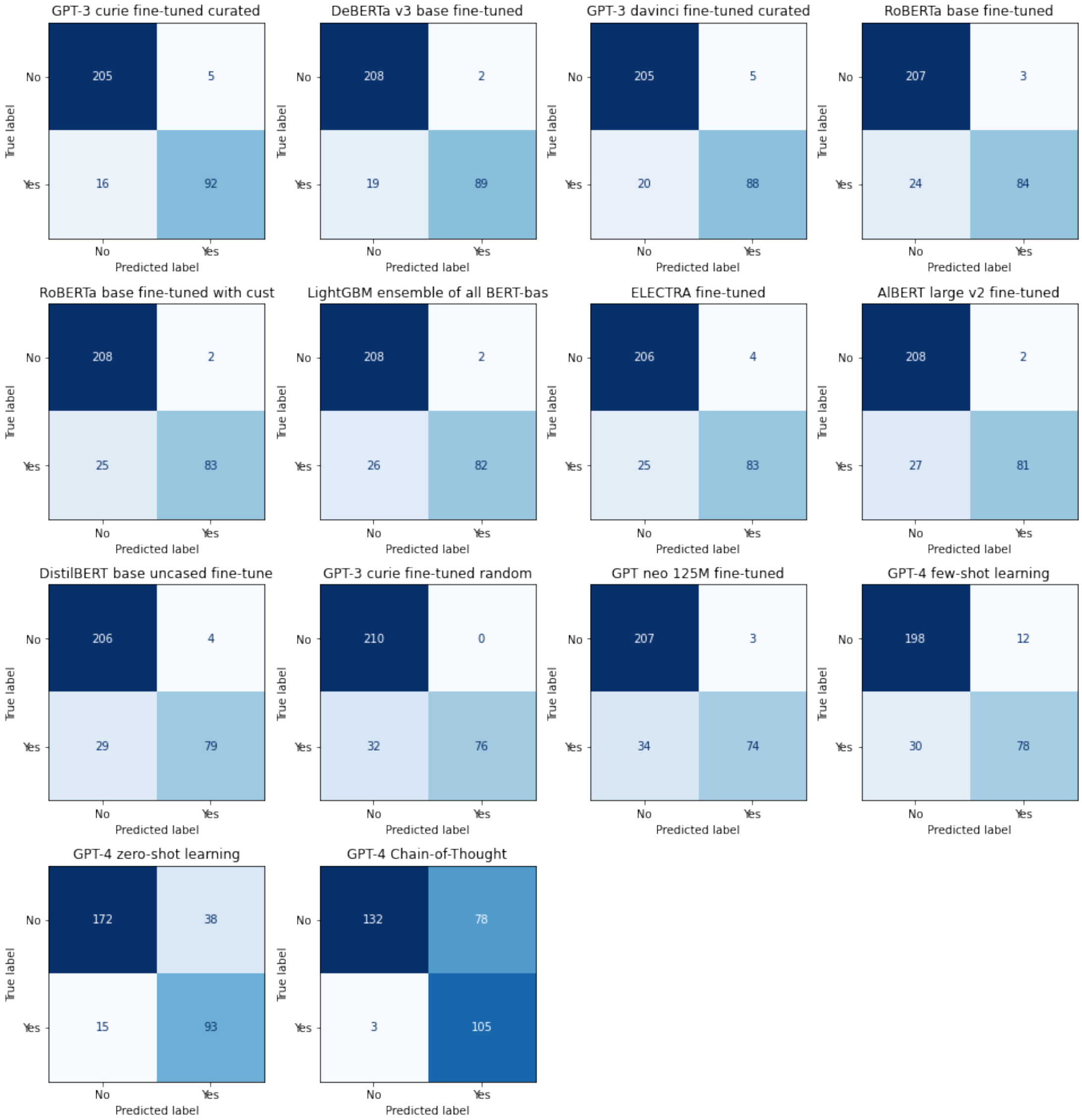


Figure 6: Confusion matrices of the models

5. Perspectives for future work

The experiments conducted for the CheckThat! 2023 competition have demonstrated the potential of GPT and BERT models in the task of check-worthiness. However, there are several avenues for future research that could further improve the performance of these models. Some of the most promising directions are discussed below.

Further dataset optimizations could be performed to improve the quality of the training data. It is not unlikely that DeBERTa v3 could benefit from reducing the train dataset by removing the most ambiguous examples. Data enrichment techniques, such as incorporating named entity recognition models, can also play its role.

Additionally, as mentioned in the section about the BERT models, we were limited by GPU resources – more specifically the memory limitation of 11GB. We could not fine-tune larger language models that could provide more accurate predictions. Various methods to reduce memory usage were attempted, but they either resulted in lower performance in terms of F1 score or were unacceptably slow. There is a decent group of larger models derived from Pythia[13], which have not been examined yet. Looking at their performance in comparison to much larger models, like GPT-4, they hold the potential for significant enhancements.

Moreover, the few-shot learning and zero-shot learning approaches could be further explored by experimenting with different models and prompts. Specifically Chain-of-Thought and Tree-of-Thoughts [36] approaches could lead to solutions that generalize better in unknown or quickly evolving domains.

Acknowledgments

The research is supported by the project "OpenFact – artificial intelligence tools for verification of veracity of information sources and fake news detection" (INFOSTRATEG-I/0035/2021-00), granted within the INFOSTRATEG I program of the National Center for Research and Development, under the topic: Verifying information sources and detecting fake news.

References

[1] C. López-Marcos, P. Vicente-Fernández, Fact checkers facing fake news and disinformation in the digital age: A comparative analysis between spain and united kingdom, Publications 9 (2021) 36.

[2] S. Cazalens, J. Leblay, P. Lamarre, I. Manolescu, X. Tannier, Computational fact checking: a content management perspective, Proceedings of the VLDB Endowment (PVLDB) 11 (2018) 2110–2113.

[3] N. Micallef, V. Armacost, N. Memon, S. Patil, True or false: Studying the work practices of professional fact-checkers, Proceedings of the ACM on Human-Computer Interaction 6 (2022) 1–44.

[4] A. Barrón-Cedeño, T. Elsayed, P. Nakov, G. Da San Martino, M. Hasanain, R. Suwaileh, F. Haouari, N. Babulkov, B. Hamdan, A. Nikolov, et al., Overview of checkthat! 2020: Automatic identification and verification of claims in social media, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings 11, Springer, 2020, pp. 215–236.

[5] F. Alam, A. Barrón-Cedeño, G. S. Cheema, S. Hakimov, M. Hasanain, C. Li, R. Míguez, H. Mubarak, G. K. Shahi, W. Zaghouani, P. Nakov, Overview of the CLEF-2023 CheckThat! lab task 1 on check-worthiness in multimodal and multigenre content, ????

[13] https://github.com/EleutherAI/pythia

[6] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, M. Kutlu, W. Zaghouani, C. Li, S. Shaar, H. Mubarak, A. Nikolov, Overview of the clef-2022 checkthat! lab task 1 on identifying relevant claims in tweets, CLEF '2022, Bologna, Italy, 2022.

[7] A. Savchev, Ai rational at checkthat! 2022: using transformer models for tweet classification, in: Working Notes of CLEF 2022–Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[8] R. Buliga Nicu, Zorros at checkthat! 2022: ensemble model for identifying relevant claims in tweets, in: Working Notes of CLEF 2022–Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[9] S. Agrestia, A. Hashemianb, M. Carmanc, Polimi-flatearthers at checkthat! 2022: Gpt-3 applied to claim detection, in: Working Notes of CLEF 2022–Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[10] R. Frick, I. Vogel, I. Nunes Grieser, Fraunhofer sit at checkthat! 2022: semi-supervised ensemble classification for detecting check-worthy tweets, in: Working Notes of CLEF 2022–Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[11] A. Eyuboglu, M. Arslan, E. Sonmezer, M. Kutlu, Tobb etu at checkthat! 2022: detecting attention-worthy and harmful tweets and check-worthy claims, in: Working Notes of CLEF 2022–Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[12] S. Mingzhe Du, S. D. Gollapalli, Nus-ids at checkthat! 2022: identifying check-worthiness of tweets using checkthat5, in: Working Notes of CLEF 2022–Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

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

[14] S. Agrestia, A. S. Hashemianb, M. J. Carmanc, PoliMi-FlatEarthers at CheckThat! 2022: GPT-3 applied to claim detection, in: Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, CLEF '2022, Bologna, Italy, 2022.

[15] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Alpaca: A strong, replicable instruction-following model, Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html 3 (2023) 7.

[16] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971.

[17] OpenAI, Improving language understanding with unsupervised learning, 2018.

[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, 2019.

[19] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.

[20] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[21] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, 2020. arXiv:2001.08361.

[22] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022. arXiv:2203.02155.

[23] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023. arXiv:2201.11903.

[24] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2020. arXiv:1910.01108.

[25] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with Disentangled Attention, 2021. arXiv:2006.03654.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019. arXiv:1907.11692.

[27] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-lingual Representation Learning at Scale, 2020. arXiv:1911.02116.

[28] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, 2020. arXiv:1909.11942.

[29] H. W. Chung, T. Févry, H. Tsai, M. Johnson, S. Ruder, Rethinking embedding coupling in pre-trained language models, 2020. arXiv:2010.12821.

[30] L. Martin, B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, B. Sagot, CamemBERT: a Tasty French Language Model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7203–7219. doi:10.18653/v1/2020.acl-main.645.

[31] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, 2020. arXiv:2003.10555.

[32] S. Black, L. Gao, P. Wang, C. Leahy, S. Biderman, GPT-Neo: Large scale autoregressive language modeling with meshtensorflow, 2021. URL: https://doi.org/10.5281/zenodo.5551208. doi:10.5281/zenodo.5551208.

[33] Z. Zeng, Y. Xiong, S. N. Ravi, S. Acharya, G. Fung, V. Singh, You Only Sample (Almost) Once: Linear Cost Self-Attention Via Bernoulli Sampling, 2021. arXiv:2111.09714.

[34] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, Y. Artzi, Revisiting few-sample bert fine-tuning, 2020. arXiv:2006.05987.

[35] P. Nakov, G. Da San Martino, T. Elsayed, A. Barrón-Cedeño, R. Míguez, S. Shaar, F. Alam, F. Haouari, M. Hasanain, W. Mansour, et al., Overview of the clef–2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings 12, Springer, 2021, pp. 264–291.

[36] S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, K. Narasimhan, Tree of thoughts: Deliberate problem solving with large language models, 2023. arXiv:2305.10601.