TurQUaz at CheckThat! 2024: A Hybrid Approach of Fine-Tuning and In-Context Learning for Check-Worthiness Estimation*

Mehmet Eren Bulut[1,*,†], Kaan Efe Keleş[1,†] and Mucahid Kutlu[2]

[1]Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Türkiye
[2]Department of Computer Science and Engineering, Qatar University, Doha, Qatar

Abstract

This paper presents our participation in the CLEF2024 CheckThat! Lab's Task-1 which focuses on determining whether passages from tweets or transcriptions are check-worthy. Task 1 covers three languages including English, Arabic, and Dutch. We propose utilizing several different instruct-tuned large language models (LLM) and aggregating their results for the Dutch dataset. In English and Arabic datasets, in addition to LLMs, we also use a fine-tuned XLM-R classifier. Our proposed method is ranked first in the Dutch dataset, fourth in the Arabic dataset, and eleventh in the English dataset.

Keywords

LLM, In Context Learning, Prompt Engineering, Check-Worthiness

1. Introduction

To combat misinformation, fact-checking websites like Snopes[1] verify claims spread on Internet and share their findings with their readers. However, this process is slow, taking about a day per claim [1]. Furthermore, false news spread much faster than true news [2], increasing the pressure on fact-checkers. Therefore, automated systems to help fact-checkers and increase their efficiency and effectiveness are highly needed.

Within the scope of fight against misinformation, we participate in Task 1 [3] of the CLEF2024 CheckThat! Lab [4]. This task involves identifying check-worthy claims within tweets or transcriptions in three languages: Arabic, Dutch, and English. Check-worthiness analysis is the first step for fact-checking systems. Despite its subjective nature [5], check-worthiness can be determined by answering questions about the content [6].

In this work, we propose a two-step process to detect check-worthy claims: an initial prediction using a fine-tuned XLM-R based classifier, followed by a more focused analysis using In-Context Learning (ICL) with various instruct-tuned LLMs. For English, we combine the classifier's predictions with those from the instruct-tuned LLMs using an F1-weighted averaging method. In the Arabic track, we aggregate predictions from the same sources (i.e., fine-tuned classifier and instruct-tuned LLMs) via majority voting. Finally, for Dutch, we rely solely on ICL with the instruct-tuned LLMs, aggregating their predictions using majority voting to produce the final label. In the official ranking, we are ranked first (out of 16) in the Dutch dataset, fourth (out of 14) in the Arabic dataset, and eleventh (out of 27) in the English dataset.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
*Corresponding author.
†These authors contributed equally.
mehmeterenbulut@etu.edu.tr (M. E. Bulut); kaanefekeles@etu.edu.tr (K. E. Keleş); mucahidkutlu@qu.edu.qa (M. Kutlu)

[1]https://www.snopes.com

2. Related Work

The concept of check-worthiness is thoroughly explained in recent studies [6]. Matwin et al. [7] approach the problem as a 3-class classification task, categorizing sentences as not check-worthy, unimportant check-worthy, and important check-worthy, thereby distinguishing between irrelevant and valuable check-worthy sentences. However, several works define the problem as a binary classification [8] as in this lab or a ranking problem [9].

In the early studies on check-worthiness, researchers explored various features such as named entities [10] and syntactic dependency tags [11]. Gencheva et al. [12] study the contextual cues that might indicate the check-worthiness of a sentence in transcripts. They report that the duration of the speech and the presence of accusations against political opponents are correlated with check-worthiness. However, Hansen et al. [13] discuss the flaws of using hand-crafted features for check-worthy claim detection.

In recent years, several studies report effectiveness of transformer models in detecting check-worthy claims, exploring various data engineering methods such as cross-lingual training [14], generating data using LLMs [15], and contextually sensitive lexical augmentation [16]. With the recent developments in generative models, researchers also explored their impact on detecting check-worthy claims. For instance, Sawinski et al. [17] conducted a comparative study of GPT and BERT models for the detection of check-worthy claims. Their findings indicate that fine-tuned BERT models can perform comparably to large language models such as GPT-3 in identifying check-worthy claims, demonstrating that both models have significant potential for automated fact-checking systems. In our work, we utilize multiple LLMs and with in context learning and a fine-tuned XLM-R model and aggregate their results to reach a final decision.

3. Proposed Approach

For each language, we develop a slightly different method. In the Arabic and English tracks, we propose a two-stage approach to determine check-worthy statements. Our method combines a fine-tuned XLM-R classifier with in-context learning (ICL) using multiple different instruct-tuned models. The aggregation method varies between the Arabic and English datasets. For the Dutch dataset, we opted to exclude the fine-tuned classifier, relying solely on in-context learning due to the time constraints of the lab.

Our two-stage approach aims to improve the prediction performance by combining the classification effectiveness of fine-tuned models with the natural language understanding capabilities of instruct-tuned LLMs. Firstly, we fine-tune an XLM-R model using the training dataset. Predictions with high confidence scores are likely to be correct while those with low confidence scores can be considered nearly random labeling. **Figure 1** shows the distribution of correctly and incorrectly classified cases for the confidence scores of our fine-tuned XLM-R model. We observe that the classifier achieves an average confidence score of 0.94 for correct classifications, in contrast to an average confidence score of 0.74 for incorrect classifications. To increase the effectiveness of our approach for these challenging examples, they are passed to our ICL labeler. Here, we devise a specific prompt to directly query multiple instruct-tuned models, asking whether a given sample is check-worthy. We then aggregate these models' outputs to determine if a sample is check-worthy or not.

Now we explain the details of our ICL labeler (Section 3.1) and the differences in our approach across languages (Section 3.2).

3.1. Labeling with In-Context Learning

In-context learning is a technique where an LLM is prompted to solve a task at inference time without updating its weights. This is achieved through a carefully curated prompt that includes explanations or examples of the task. The concept was introduced and defined by Brown et al. [18]. During unsupervised pre-training, a language model develops a broad set of skills and pattern recognition

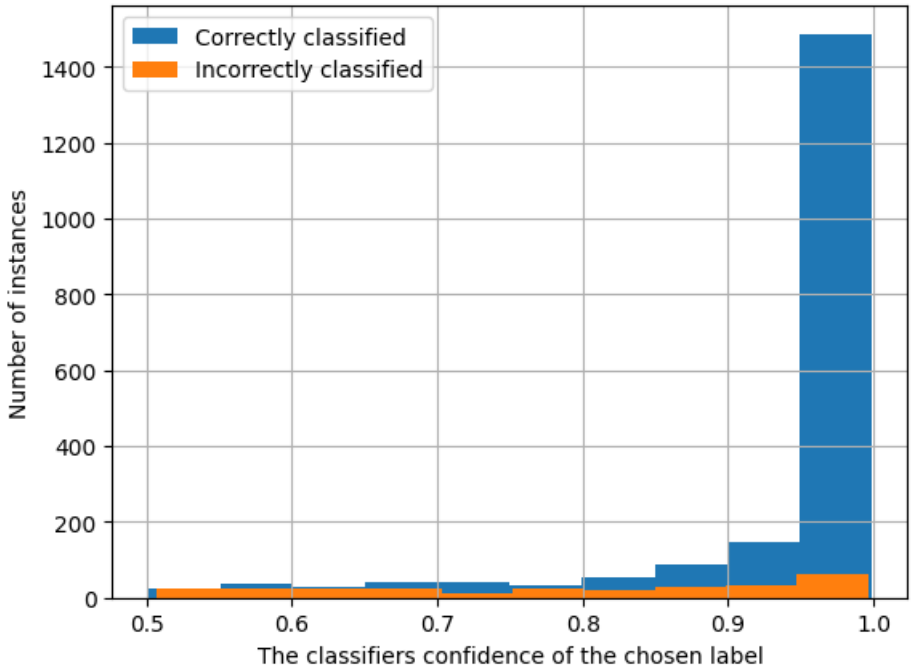

Figure 1: The accompanying graph illustrates the distribution of the classifier's confidence scores (defined as the probability assigned to the chosen label) for correctly and incorrectly classified instances on the English dataset. The x-axis represents the confidence scores, while the y-axis indicates the number of instances. Correctly classified instances are represented by blue bars, whereas incorrectly classified instances are depicted in orange. For this plot, the training data for English subtask was divided into a 9:1 train-test split, and the reported results were obtained by fine-tuning on the training set and predicting on the test set.

abilities. At inference time, the model uses these abilities to quickly adapt to or recognize the desired task. The term "in-context learning" refers to this adaptive process, which occurs within the forward pass of each sequence.

The prompt developed for this task, illustrated in **Figure 2**, consists of three distinct sections. The first section provides an explanation of the task, emphasizing the importance of accurate information. In the second section, explicit instructions are given to the model to generate a data in JSON format with three specific labels: "candidate_text," "reasoning," and "label." The requirements for each label are clearly defined. The final section elaborates on the content to be included under the "reasoning" tag and the candidate passage to be validated for check-worthiness, stressing adherence to the JSON format.

3.2. Approaches for Different Languages

In this section, we explain our specific approaches for each language.

3.2.1. English

Firstly, we identify the samples to be passed to the ICL labeler. We refer this subset of the data *ICL subset*, which includes instances where the confidence score of the fine-tuned classifier is below 90%. We select 90% as threshold because the performance of the model noticeably decreases when its confidence score is below 90% as seen in Figure 1. This subset is then processed by the ICL labeler. Finally, we combine the labels from the fine-tuned classifier and the ICL labeler with a weighted averaging aggregation, where the weights are determined by their F1 scores. The details of this aggregation method are as follows.

Weighted Average Aggregation with F1 Scores. To determine the labels for the samples in the ICL subset, we calculate the F1 score for the XLM-R model and the instruct-tuned LLMs on the ICL subset of

As an expert classifier, you're tasked with determining the check-worthiness of claims in tweets or transcriptions. In this task, your goal is to decide whether a given claim is worth fact-checking. This involves assessing factors such as verifiability, potential harm, and factual accuracy.

Please generate a JSON file with the following fields:
1. "candidate_text": The text of the claim from the tweet or transcription.
2. "reasoning": An explanation or reasoning supporting your decision.
3. "label": The label indicating whether the claim is worth fact-checking. Please assign one of the following values:
  - 1: If the claim is worth fact-checking.
  - 0: If the claim is not worth fact-checking.

Ensure that the JSON file adheres to the following format:

```
{
 "candidate_text": "The claim text goes here.",
 "reasoning": "The reasoning goes here.",
 "label": 0 or 1 depending on the claim's check-worthiness.
}
```

For the "reasoning" field, you must provide evidence or reasoning supporting your decision. Consider factors such as verifiability, potential harm, and factual accuracy in your assessment.

Remember, your task is to provide clear and concise justifications for your label assignment, leveraging your expertise as a classifier to make informed decisions.

Adhere strictly to the JSON format. Do not include ANY OTHER text.

Here is the candidate passage: **the actual candidate passage**

Expected Raw Output From The Model

{'candidate_text': **the actual candidate passage**,
'reasoning': "This claim is worth fact-checking because ...",
'label': 1}

Figure 2: Prompt Instructions Provided to the Model

the training set. Afterwards, we compute a weighted average of their output labels, using the F1 scores as weights. Samples are labeled as check-worthy if this weighted average exceeds a hyperparameter α. The F1 scores for the XLM-R model and the instruct-tuned LLMs, along with the hyperparameter α are determined during the development phase and used during testing.

3.2.2. Arabic

For the Arabic task, we aggregate all labels from both the fine-tuned classifier[2] and the ICL labels. We employ a super majority voting system for our aggregation strategy, requiring agreement from four out of five label sources for a label to be selected. We label the cases we do not reach this threshold, as *not-check-worthy*.

3.2.3. Dutch

For the Dutch task, we rely exclusively on the ICL approach due to the time constraints of the lab. We aggregate the predictions of LLMs based on majority voting to reach a final decision.

[2]https://huggingface.co/keles/clef1ar

4. Experiments

4.1. Implementation Details

For our pre-trained classifier we used the multilingual "FacebookAI/xlm-roberta-large" [19] model[3] as our base pretrained model. For in-context learning, we employed available APIs for GPT-3.5, GPT-4[4], and Gemini 1 Pro[5]. We also used open-source models including Meta-Llama-3-8B-Instruct[6][7] and Mistral-7B-Instruct-v0.2[8]. We quantize these open-source models down to 4-bit precision to accommodate the hardware limitations. We used HuggingFace's [20] `generate` API for text generation. **Table 1** shows the specific models we used for each task. We were not able to use some of the models in Arabic and Dutch due to quota limits and time constraints.

Table 1
Models used for each language.

Models	RoBERTa	Mistral 7Bv2	Llama3 8B	GPT-3.5	GPT-4	Gemini-1.0-pro
English	✓	✓	✓	✓	✓	✓
Arabic	✓	✓	✓	✓	✓	✗
Dutch	✗	✓	✓	✓	✗	✗

As our transformer model we used "FacebookAI/xlm-roberta-large," for both English and Arabic languages. We fine-tuned models for each language separately using HuggingFace's Trainer API[9]. We set the same parameters for both languages: a batch size of 16, a learning rate of 3×10^{-5}, and 5 epochs.

For the Arabic fine-tuned model, we performed evaluations in every 200 steps using the test partition and calculated F_1 score at each interval to monitor the model's performance. The optimal performance was identified at 2.5 epochs, which corresponds to the $1{,}000^{th}$ step (out of 2,065 steps). Beyond this point, additional training did not result in any further improvement in the F_1 score on the test set. Thus, we selected this checkpoint for the remainder of the analysis in this study.

For the English fine-tuned model, evaluations were carried out in every 500 steps. The model checkpoint with the lowest test set loss was chosen for further analysis, which was observed at the 2500^{th} step (out of 6,330 steps). Further configuration details and the fine-tuned models we used are available at Huggingface[10,11].

4.2. Dataset

The dataset[12] consists of passages derived from tweets or transcriptions. Each passage is annotated with a binary label indicating its checkworthiness. **Table** 2 presents statistics about the data. For the experiments conducted during the development phase of the lab, the training dataset shared by the organizers of the lab was split into 80% for training and 20% for testing for Arabic while 90% of the English training dataset was allocated for training and 10% for testing.

[3]https://huggingface.co/FacebookAI/xlm-roberta-large
[4]https://openai.com/index/introducing-chatgpt-and-whisper-apis/
[5]https://ai.google.dev/gemini-api/docs/api-overview
[6]https://ai.meta.com/blog/meta-llama-3/
[7]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[8]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[9]https://huggingface.co/docs/transformers/en/main_classes/trainer
[10]https://huggingface.co/keles/clef1ar
[11]https://huggingface.co/keles/clef1eng
[12]https://gitlab.com/checkthat_lab/clef2024-checkthat-lab/-/tree/main/task1/data

Table 2
Statistics about the dataset used in our study. CW: Check-Worthy

		English	Arabic	Dutch
Train	Count	22501	7333	995
	%CW	%24	%30	%40
Test	Count	341	610	1000
	%CW	%25	%35	%39

4.3. Experimental Results

4.3.1. Results on the English Dataset

We used the 10% of the training dataset for testing purposes during the development phase, as mentioned before. We call this subset as *evaluation set* throughout the paper. In our experiments with English dataset, we evaluate the impact of our ICL labeler with different aggregation methods. In particular, we compare the performance of three methods: i) the fine-tuned classifier, ii) fined tuned classifier and ICL labeler with majority voting, and iii) fined tuned classifier and ICL labeler with F_1 weighted averaging. **Table 3** shows F_1 scores on both evaluation and test sets.

Table 3
F_1 Scores of Our Methods for English

Model	Evaluation Set	Test Set
XLM-R	**0.768**	**0.732**
XLM-R & ICL with Majority Voting	0.753	0.716
XLM-R & ICL with F_1 Weighted Averaging	0.767	0.718

The fine-tuned XLM-R achieves the highest F_1 score, showing that ICL has negative impact on the overall performance. Among the methods that use ICL, F_1 weighted averaging yields a higher score than the majority voting, highlighting the importance of utilizing sophisticated aggregation techniques.

4.3.2. Results on the Arabic Dataset

Table 4 presents the F_1 scores for each model and results for the aggregated results under both majority and super-majority voting methods for Arabic dataset. As explained in 3.2.2, for the Arabic task, we employ super majority voting, which requires four out of five sources to label a claim as check-worthy. However, as illustrated in Table 4, a basic majority voting approach yields a higher F_1 score. In contrast to our results for English, aggregation improves the F_1 scores in the Arabic dataset, as both aggregation approaches outperform all other models.

Table 4
F_1 Scores for Various Methods on Arabic Dataset.

XLM-R	Mistral 7Bv2	Llama3 8B	GPT-3.5	GPT-4	Majority Voting	Super-Majority Voting
0.47	0.42	0.47	0.47	0.52	**0.552**	0.532

4.3.3. Results on the Dutch Dataset

As mentioned in 3.2.3, for the Dutch task, we employ a straightforward in-context learning approach with label aggregation. We use three models and aggregate their individual predictions via majority voting. In this experiment, we also assess the impact of inclusive aggregation in which a claim is labeled as check-worthy if at least one of the models predicted as check-worthy. **Table 5** presents the

results of this method on the Dutch training data. We observe that the inclusive aggregation yields the highest performance. In addition, LLama 3 and GPT3.5 achieve higher scores than the majority voting aggregation. This might be because of the low performance of Mistral 7Bv2.

Table 5
Comparative F1 Scores for Dutch Task Models

Model	Mistral 7Bv2	Llama3 8B	GPT-3.5	Inclusive Agg.	Majority Voting Agg.
F1 Score	0.310	0.554	0.580	0.587	0.543

4.4. Official Ranking

Due to the time constraints of the lab, we had to pick the models to be submitted based on the results that we had in the development period. In particular, we selected the following configurations as our primary model: F_1 weighted averaging for English, super-majority voting for Arabic, and majority voting for Dutch. However, based on our follow-up experiments after the submission deadline, we observed that these are not the best performing configurations. Nevertheless, our primary models achieved notable success. In particular, we are ranked first (out of 16) in the Dutch track, fourth (out of 14) in the Arabic track, and eleventh (out of 27) in the English track.

4.5. Inspecting Difficult Samples

38 English sentences (out of 341) in the test set are classified incorrectly with all of our methods. After inspecting these sentences, we notice that determining whether these sentences are check-worthy is difficult even for human evaluators. Some of these samples are shown in **Table 6**. These results show that LLMs might be beneficial to detect sentences which might need label correction.

Table 6
Sample cases where all of our methods failed to classify correctly.

Sentence	**Label**
"And it's not like it was 25 years ago, it was three and three quarters."	Not Check-Worthy
"But the Biden administration sends Blinken, Yellen over there."	Not Check-Worthy
"We're skating on thin ice and we cannot set a precedent where the party in power uses police force to indict its political opponents."	Check-Worthy

5. Conclusion

In this work, we explored utilizing both fine-tuned pretrained transformers and instruct-tuned LLMs through ICL. Our proposed methods demonstrated considerable success across three languages. Our first-place ranking in the Dutch track highlights the remarkable zero-shot capabilities of current LLMs, while our results in Arabic and English underscore the potential of combining traditional fine-tuning with ICL techniques. In the future work, we plan to extend our work and explore how predictions of LLMs can be aggregated effectively.

References

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

[2] S. Vosoughi, D. Roy, S. Aral, The spread of true and false news online, science 359 (2018) 1146–1151.
[3] M. Hasanain, R. Suwaileh, S. Weering, C. Li, T. Caselli, W. Zaghouani, A. Barrón-Cedeño, P. Nakov, F. Alam, Overview of the CLEF-2024 CheckThat! lab task 1 on check-worthiness estimation of multigenre content, ????
[4] A. Barrón-Cedeño, F. Alam, T. Chakraborty, T. Elsayed, P. Nakov, P. Przybyła, J. M. Struß, F. Haouari, M. Hasanain, F. Ruggeri, X. Song, R. Suwaileh, The clef-2024 checkthat! lab: Check-worthiness, subjectivity, persuasion, roles, authorities, and adversarial robustness, in: N. Goharian, N. Tonellotto, Y. He, A. Lipani, G. McDonald, C. Macdonald, I. Ounis (Eds.), Advances in Information Retrieval, Springer Nature Switzerland, Cham, 2024, pp. 449–458.
[5] Y. S. Kartal, M. Kutlu, Trclaim-19: The first collection for turkish check-worthy claim detection with annotator rationales, in: Proceedings of the 24th Conference on Computational Natural Language Learning, 2020, pp. 386–395.
[6] F. Alam, S. Shaar, F. Dalvi, H. Sajjad, A. Nikolov, H. Mubarak, G. D. S. Martino, A. Abdelali, N. Durrani, K. Darwish, A. Al-Homaid, W. Zaghouani, T. Caselli, G. Danoe, F. Stolk, B. Bruntink, P. Nakov, Fighting the covid-19 infodemic: Modeling the perspective of journalists, fact-checkers, social media platforms, policy makers, and the society, 2021. arXiv:2005.00033.
[7] S. Matwin, S. Yu, F. Farooq, N. Hassan, F. Arslan, C. Li, M. Tremayne, Toward automated fact-checking, Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017) 1803–1812. doi:10.1145/3097983.3098131.
[8] A. Barrón-Cedeño, F. Alam, A. Galassi, G. Da San Martino, P. Nakov, T. Elsayed, D. Azizov, T. Caselli, G. S. Cheema, F. Haouari, et al., Overview of the clef–2023 checkthat! lab on checkworthiness, subjectivity, political bias, factuality, and authority of news articles and their source, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2023, pp. 251–275.
[9] P. Nakov, A. Barrón-Cedeno, T. Elsayed, R. Suwaileh, L. Màrquez, W. Zaghouani, P. Atanasova, S. Kyuchukov, G. Da San Martino, Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 9th International Conference of the CLEF Association, CLEF 2018, Avignon, France, September 10-14, 2018, Proceedings 9, Springer, 2018, pp. 372–387.
[10] K. Yasser, M. Kutlu, T. Elsayed, bigir at clef 2018: Detection and verification of check-worthy political claims., in: CLEF (Working Notes), 2018.
[11] C. Lespagnol, J. Mothe, M. Z. Ullah, Information nutritional label and word embedding to estimate information check-worthiness, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 941–944.
[12] S. U. . O. Bulgaria, P. Gencheva, P. Nakov, H. Qatar, Qatar Computing Research Institute, L. Màrquez, A. Barrón-Cedeño, I. Koychev, A context-aware approach for detecting worth-checking claims in political debates, RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning (2017) 267–276. doi:10.26615/978-954-452-049-6_037.
[13] C. Hansen, C. Hansen, S. Alstrup, J. Grue Simonsen, C. Lioma, Neural check-worthiness ranking with weak supervision: Finding sentences for fact-checking, in: Companion proceedings of the 2019 world wide web conference, 2019, pp. 994–1000.
[14] Y. S. Kartal, M. Kutlu, Re-think before you share: A comprehensive study on prioritizing check-worthy claims, IEEE transactions on computational social systems 10 (2022) 362–375.
[15] A. Modzelewski, W. Sosnowski, A. Wierzbicki, Dshacker at checkthat! 2023: Check-worthiness in multigenre and multilingual content with gpt-3.5 data augmentation, Working Notes of CLEF (2023).
[16] E. Williams, P. Rodrigues, S. Tran, Accenture at checkthat! 2021: interesting claim identification and ranking with contextually sensitive lexical training data augmentation, arXiv preprint arXiv:2107.05684 (2021).
[17] M. Sawiński, K. Węcel, E. P. Księżniak, M. Stróżyna, W. Lewoniewski, P. Stolarski, W. Abramowicz, Openfact at checkthat! 2023: head-to-head gpt vs. bert-a comparative study of transformers language models for the detection of check-worthy claims, in: CEUR Workshop Proceedings,

volume 3497, 2023.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[19] 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, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747. doi:10.18653/v1/2020.acl-main.747.

[20] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, 2020. arXiv:1910.03771.