This is not correct! Negation-aware Evaluation of Language Generation Systems

Miriam Anschütz and **Diego Miguel Lozano** and **Georg Groh**
School for Computation, Information and Technology
Technical University of Munich, Germany
{miriam.anschuetz, diego.miguel}@tum.de, grohg@in.tum.de

Abstract

Large language models underestimate the impact of negations on how much they change the meaning of a sentence. Therefore, learned evaluation metrics based on these models are insensitive to negations. In this paper, we propose *NegBLEURT*, a negation-aware version of the BLEURT evaluation metric. For that, we designed a rule-based sentence negation tool and used it to create the *CANNOT* negation evaluation dataset. Based on this dataset, we fine-tuned a sentence transformer and an evaluation metric to improve their negation sensitivity. Evaluating these models on existing benchmarks shows that our fine-tuned models outperform existing metrics on the negated sentences by far while preserving their base models' performances on other perturbations.

1 Introduction

Previous work has shown that large language models such as BERT (Devlin et al., 2019) lack understanding of negated phrases and do not attribute sufficient importance to the word "not" (Ettinger, 2020; Hosseini et al., 2021). Nevertheless, many widely-used metrics to evaluate natural language generation (NLG) systems, such as BERTScore (Zhang* et al., 2020) or BLEURT (Sellam et al., 2020), are based on these models. Automatic evaluation is indispensable when language models are published nearly every day. Moreover, large benchmark datasets make a human evaluation of language models infeasible. Therefore, metric scores are among the most important model selection criteria. However, here we see a severe issue when these metrics fail to distinguish between sentences and their negated versions. Especially when considering fact-checking or entailment prediction, an uninterpreted "not" can invalidate the entire output of the model and, thus, reduce the trustworthiness of such systems.

While there have been approaches to improve the negation-awareness in natural language in-
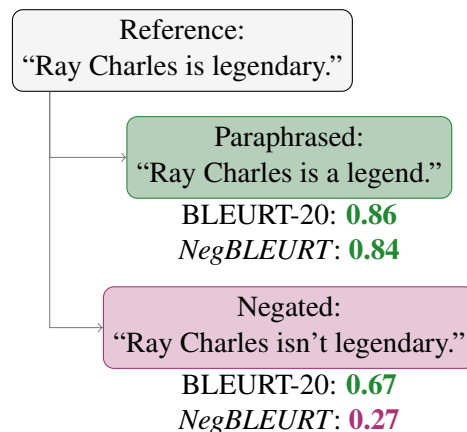

Figure 1: Existing metrics like BLEURT-20 fail to score negated sentences correctly. We propose *NegBLEURT* that overcomes this problem while preserving detection performance on other perturbations.

ference (NLI) or sentiment classification models (Moore and Barnes, 2021), the task of negation-sensitive evaluation of such systems is lacking behind (Karpinska et al., 2022). An example of such failure is shown in Figure 1, where a reference sentence is both paraphrased and negated. The well-established BLEURT-20 metric (Pu et al., 2021) gives a relatively high score of 0.67 to this negated sentence, suggesting that it does not fully capture the negation in the sentence.

To extend negation research to the topic of evaluation, we present a negation-aware version of BLEURT, named *NegBLEURT* (Figure 1). In addition, we released a negation-aware sentence transformer (Reimers and Gurevych, 2019) based on an MPNet model (Song et al., 2020) that extends the application of negation sensitivity to a broader range of tasks. Both models were fine-tuned on a labeled dataset with about 30 thousand sentence pairs in both their negated and paraphrased versions. We publish this dataset and the sentence negator used to create it together with our models. More specifically, our contributions are:

- We open-sourced a rule-based, sentence-level negation tool and released it as a Python package.

- Based on this negator, we built a Compilation of ANnotated, Negation-Oriented Text-pairs (CANNOT). This negation evaluation dataset can be used to fine-tune evaluation metrics for negation awareness or probe their sensitivity.

- We fine-tuned an MPNet model on our negation dataset. This model returns sentence embeddings that are sensitive to negations.

- We published NegBLEURT, a negation-aware version of the BLEURT evaluation metric.

Our models were evaluated on various benchmark datasets showing that they greatly outperform their base model on negated sentences while delivering similar performance on other tasks.

2 Related work

This section highlights existing work investigating the negation awareness of the BERT language model and different NLG evaluation metrics. Furthermore, we present approaches to improve this awareness with negation pre-training.

2.1 Studies on negation understanding

Ettinger (2020) showed that BERT is insensitive to negations. She designed a completion task where the hypernym description of a word was masked. In addition, a "not" was added to the sentences, resulting in affirmative and negated versions of each sentence. BERT predicted correct hypernyms for both versions, meaning that the model failed to consider the negation indicator. Similar results were achieved by Kassner and Schütze (2020). However, they obtained correct completions when the model was fine-tuned on a negation classification task.

To investigate if BERT-based metrics inherit this lack of negation awareness, Leung et al. (2022) inspected evaluation metrics such as BERTScore (Zhang* et al., 2020) and Sentence-BERT (Reimers and Gurevych, 2019). They used these metrics to calculate the semantic similarity between 20 equivalent and negated sentence pairs. The BERT-based metrics returned high similarity values, indicating they were robust to negations. More large-scale experiments were performed by Karpinska et al. (2022) and Sai et al. (2021) where different metrics, including BERTScore and BLEURT (Sellam et al.,

2020), were evaluated on a collection of sentence and word-level perturbations, including negations and antonyms. Both studies show that most suggested evaluation metrics struggle to detect negations and deviate strongly from human evaluations. Koch et al. (2022) examined the robustness towards these perturbations on a continuous scale by gradually introducing more perturbations to the sentences and, hence, decreasing their quality step by step. While the metrics' scores lowered for other perturbations, the scores for the negated sentences remained relatively high, indicating insensitivity towards negation.

2.2 Improving negation awareness

Negation awareness is crucial for the task of natural language inference (NLI), in which models predict if two sentences entail or contradict each other. Hence, multiple datasets with negated samples and models trained on them have been published (Geiger et al., 2020; Helwe et al., 2022). As such, Hosseini et al. (2021) created BERTNOT by training on the negated LAMA dataset with an unlikelihood training objective. Other negation-aware BERT models are NegBERT (Khandelwal and Sawant, 2020) and CueBERT (Truong et al., 2022a), which were trained for the task of negation cue detection and negation scope resolution. Another task that heavily relies on negation awareness is sentiment classification. Moore and Barnes (2021) proposed multi-task learning with a negation speculation auxiliary task to improve the model's performance on negated samples.

While there has been extensive work on negation understanding in NLI and other tasks, we could not find approaches to improve negation awareness for NLG evaluation metrics. This paper tries to close this gap by pre-training metrics on negated sentences.

3 Contrastive negation dataset

To make an evaluation metric aware of negations, we need a dataset containing pairs of reference and candidate sentences and a label of how well the candidate fits the reference. To have a balanced dataset, we not only need negated, but also meaning-preserving paraphrases of the reference sentence. As described in the previous section, there exist multiple datasets focusing on negations. However, most of these datasets are either targeted towards specific tasks such as NLI or only contain

negated sentence pairs. Thus, we processed and aggregated the existent datasets producing a Compilation of ANnotated, Negation-Oriented Text-pairs (CANNOT), which addresses and solves these issues. More specifically, our negation-evaluation dataset is based on the following resources:

- *Not another Negation Benchmark* (Truong et al., 2022b): This dataset was published to improve negation awareness in NLI and includes negated sentence pairs. We filtered for samples with the label "contradiction", resulting in 117 negated pairs.

- *Automated Fact-Checking of Claims from Wikipedia* (Sathe et al., 2020): This dataset contains claim-refutation pairs from texts extracted from Wikipedia. The refutation, i.e., the factually incorrect sentence, is often created by negating the claim or replacing one of its words with an antonym. Including more nuanced negations as antonyms and other semantic artifacts diversify the negations in our dataset, making the models trained on it more robust to different negation forms. Nonetheless, many refuted sentences also included further augmentations, such as hallucinations. To only focus on negations, we discarded sentence pairs that had a Jaccard similarity coefficient of less than 0.55 or differed in length by four or more words. The word splits were obtained with simple white-space tokenization. After the processing, $14,970$ samples were kept.

- *GLUE Diagnostic Dataset* (Wang et al., 2018): Again, this dataset is targeted to NLI and contains changes beyond pure negation. As with the other datasets, we selected only samples labeled as contradiction and dropped pairs with low Jaccard similarity coefficients or large differences in their lengths. This selection resulted in 154 samples.

- *Sentiment-annotated reviews* (Kotzias et al., 2015): This dataset contains online reviews with a strong positive or negative sentiment and, thus, broadens the domains covered by our data. We selected sentences with an auxiliary verb and at most 33 words. Then, we used our rule-based negation tool (see following subsection 3.1) to create negated versions

of the sentences. In total, $2,110$ further samples were collected.

This resulted in a dataset with negated sentence pairs. To extend it with meaning-preserving paraphrases, we used a PEGASUS model pre-trained for this task [1] and created paraphrased versions of each of the references. Finally, the dataset was augmented by adding a swapped version of each pair. This results in a dataset of $68,780$ sentence pairs with equal distribution of negated and equivalent samples. The pre-processed versions of the underlying datasets and our resulting dataset are publicly available on GitHub [2].

3.1 Rule-based sentence negator

While previous work used rule-based negation to create negation datasets before, their negators are often not open-source or lost in large repositories with code for the overall goal of the project. Therefore, we publish a lightweight and open-source sentence negation tool as Python module[3] that can be used beyond the scope of this paper.

Our negation tool focuses on verbal negations and supports the addition and deletion of negation cues on a sentence level. The flowchart for the negator is shown in Figure 2, accompanied by example sentences in Table 1. To determine whether a sentence is negated and to distinguish between auxiliary verb forms and common verbs, we first apply the POS tagger provided by the spaCy package (Honnibal et al., 2020). A negated sentence is a sentence where a token in the dependency tree is labeled as "neg" (branches (1) and (2) in Figure 2). We differentiate between the auxiliary "do" and other auxiliary verbs to remove this negation particle. We either entirely remove the negated "do" (e.g., *don't like* → *like*) or remove the negation particle from the auxiliary (e.g., *isn't / is not* → *is*). Afterward, the remaining verb is conjugated to match the form of the auxiliary[4].

To negate an affirmative sentence (branches (3)-(5) in Figure 2), we extract the root verb of the dependency tree. If this verb is a full verb and not an auxiliary, we add a negated "do" matching the
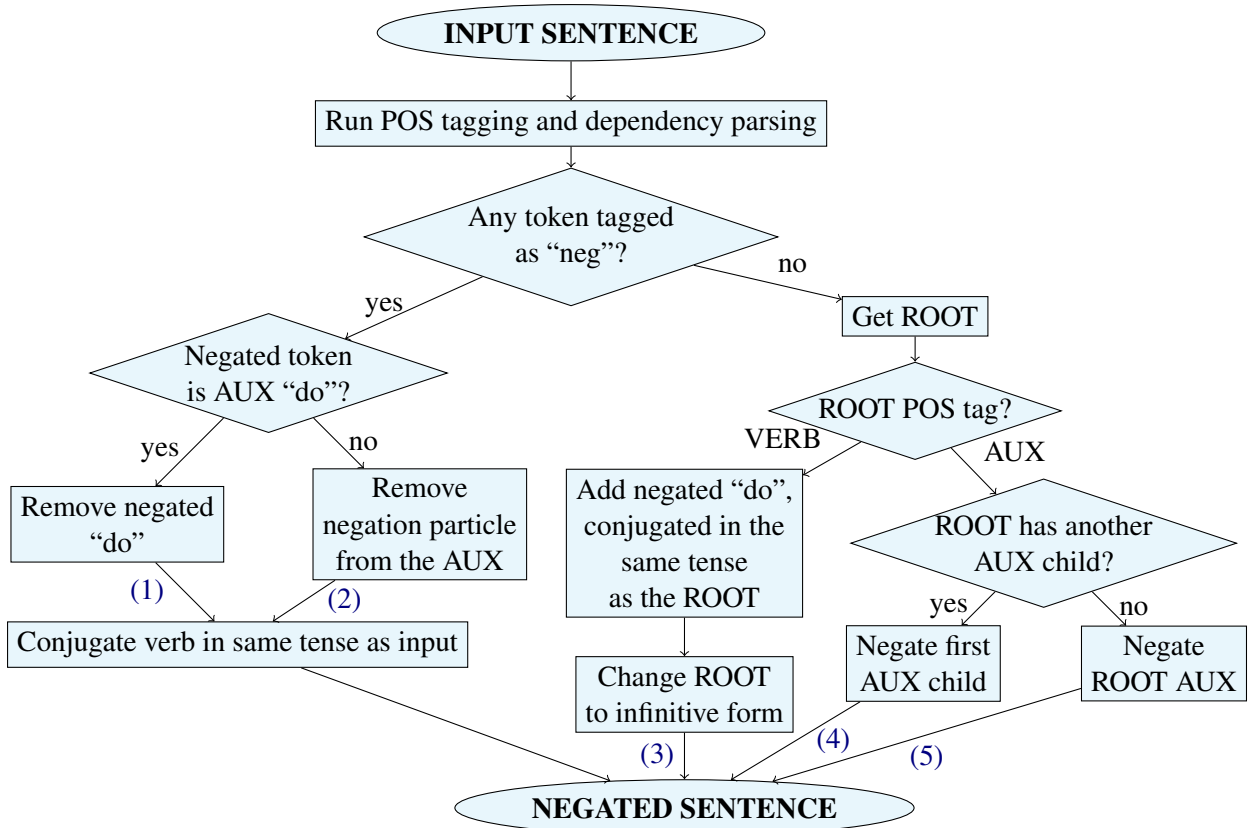

Figure 2: Flow chart for rule-based sentence negation. The negator can delete negation cues from already negated sentences as well as add them to negate a sentence.

Branch	Input sentence	Negated sentence
(1)	I *didn't know* what to do.	I *knew* what to do.
(2)	I *have never been* to Paris.	I *have been* to Paris.
(3)	I *enjoyed* it so much.	I *did not enjoy* it so much.
(4)	I *will be* there.	I *won't be* there.
(5)	I*'m* very hungry.	I*'m not* very hungry.

Table 1: Example sentences for different branches in our rule-based negator. Examples (1) and (2) remove a negation from the sentence while examples (3)-(5) add one. The user can decide whether the system should prefer contracted versions like "won't" instead of "will not".

conjugation of the respective verb. If the root is an auxiliary verb, we either negate its first auxiliary child, if any, or otherwise negate the auxiliary itself. The user can decide if the negator should prefer the contracted version, e.g., "don't", or write all words separately, e.g., "do not".

3.2 NLG evaluation dataset

We aimed to make evaluation metrics more sensitive towards negations while preserving their ability to detect other errors. Therefore, we added data from the WMT Metrics Shared Task (Bojar et al., 2017) to our dataset. This human-annotated data focuses on common errors in machine translation outputs and was used to train multiple evaluation metrics before (Sellam et al., 2020). We limited ourselves to the datasets from the years 2015 to 2017 since, upon manual review, the more recent datasets were noisier and contained misannotations. We filtered for samples with a score above -1, resulting in $9,264$ labeled samples. Most of the scores range between 0 and 1, where a score of 1 or higher indicates a perfect paraphrase. Our negation data was labeled with a score 0 for the negated pairs and a score 1 for the meaning-preserving pairs to match the annotation schema of the WMT

data. Finally, we split both the WMT data and our negation dataset into training, evaluation, and test subsets with a ratio of 80:10:10 and combined the respective subsets. This results in training data with $62,435$ samples and test data with $7,804$ samples, with a 12%-88% distribution of WMT and negation data both.

4 Negation-aware models

We publish two different models fine-tuned on our CANNOT data. On the one hand, we fine-tuned a sentence transformer to return negation-aware sentence embeddings. While the cosine similarity of two embeddings can be applied as an evaluation metric, the embedding representations have broader use cases, e.g., for topic modeling (Reimers and Gurevych, 2019). On the other hand, we fine-tuned BLEURT (Sellam et al., 2020), explicitly targeted towards the evaluation task. Both models are published in our GitHub repository[5] and on the Hugging Face Model Hub[6]. Our NegBLEURT can also be utilized within the Hugging Face Evaluate library (Wolf et al., 2020)[7].

4.1 Sentence Transformer fine-tuning

Our negation sentence encoder is based on an `all-mpnet-base-v2`[8] model and fine-tuned with the Sentence Transformer library (Reimers and Gurevych, 2019). We trained on our negation training data for one epoch with a batch size of 64, a learning rate of $2e^{-5}$, and an AdamW optimizer. We utilized a multiple negatives ranking loss to increase the latent distance between correctly paraphrased and negated samples. To create an evaluation metric based on this sentence transformer, embeddings for both the reference and candidate sentence are computed and then scored by their cosine similarity. We call this model the negated MPNet, *NegMPNet*. The cosine-similarity metric based on this model achieves a Spearman correlation of 0.72 with the ratings in the CANNOT-WMT test set.

4.2 Negation aware evaluation metric

BLEURT (Sellam et al., 2020) is a reference-based NLG evaluation metric that encodes the references and candidates with a BERT model (Devlin et al., 2019) and predicts a quality score between 0 and 1 with a linear regression layer on top of the BERT model. After pre-training with augmented Wikipedia data, BLEURT was trained on WMT data (Bojar et al., 2017). We chose a BLEURT (Sellam et al., 2020) metric as the base for our evaluation metric and selected the `bleurt_tiny` checkpoint, published as test checkpoint on the official GitHub page[9]. This checkpoint is very lightweight with a hidden size of only 128, instead of 768 as in standard BERT models. We used the fine-tuning script provided by the authors, and thus, their original hyperparameters and regression (L2) loss. We fine-tuned on the CANNOT training data for 500 steps, resulting in our final *NegBLEURT* checkpoint. This model has a Spearman correlation of 0.65 with the scores of our test set.

5 Evaluation

A common issue with fine-tuning is catastrophic forgetting (Goodfellow et al., 2013), i.e., the models forget their initial knowledge and overfit the new task. Our fine-tuning approach is successful if it improves negation awareness while retaining performance on other tasks, e.g., not corrupting the detection quality of other errors in candidate sentences. We test our models on common embedding and evaluation benchmarks to test our approach and compare their performances against their respective base models.

5.1 Massive Text Embedding Benchmark (MTEB)

MTEB (Muennighoff et al., 2023) is an embedding benchmark that evaluates embeddings on multiple tasks such as classification, clustering, and semantic textual similarity (STS). It is one of the most extensive collections of tasks, and thus, we evaluated our NegMPNet on this benchmark. As our work targets English, we only evaluate on the English version of the benchmark. The results averaged per task, and the overall macro average are presented in Table 2, while the performances on the single datasets are provided in Appendix A. Unfortunately, some of the datasets (one reranking
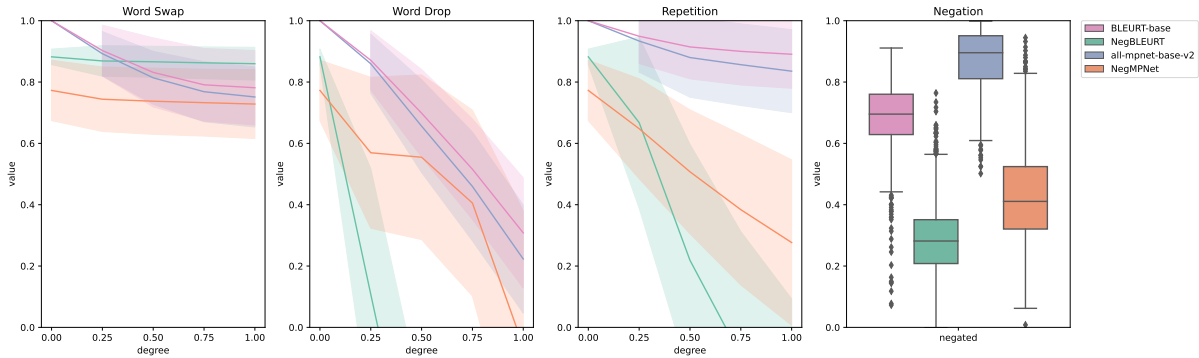
Figure 3: Metrics sensitivities to different degrees of perturbation impairment as introduced by Koch et al. (2022). Both our proposed models match the performance of their base models on the word swap, word drop and repetition perturbations but clearly outperform them on the negation detection task.

and seven retrieval) returned errors, and hence, we excluded them from our evaluation. We copied the scores for all-mpnet-base-v2 from the official leaderboard[10]. NegMPNet outperforms its base model in the classification and summarization tasks but shows a decreased performance for clustering, pair classification, and retrieval. When averaging the performances among all tasks, both models perform equally. This benchmark is not targeted towards negation, and, therefore, the results indicate that fine-tuning on our negation data does not harm NegMPNet's general embedding quality.

Benchmark (num datasets)	Model	
	all-mpnet-base-v2	NegMPNet
Average (60)	58.78	57.16
Classification (12)	65.07	70.83
Clustering (11)	43.69	38.45
Pair Classification (3)	83.04	79.05
Retrieval (20)	43.10	36.12
Reranking (3)	68.83	68.24
STS (10)	80.28	77.58
Summarization (1)	27.49	29.84

Table 2: Comparison of NegMPNet and its base model on the Massive Text Embedding Benchmark (MTEB). We evaluate on different task categories and macro average the scores.

5.2 Improved negation awareness

In this section, we analyze the improved negation awareness beyond the performance on the CANNOT-WMT test set. We selected two NLG evaluation benchmarks that probe negation sensitivity in different metrics and tested NegMPNet (with cosine similarity) and NegBLEURT on them. The results are presented in the following sections.

5.2.1 Metrics Comparison benchmark

Koch et al. (2022) probed current, learned metrics for their sensitivity to word swap, repetition, certain word drops, and negations. They gradually increased the level of impairment for all perturbations except negation to measure if the metrics could reflect upon this gradual deterioration. We evaluated our two metrics on their codebase and report the results in Figure 3. NegBLEURT matches the performance of its BLEURT base model (Sellam et al., 2020) and is sensitive to word drops and repetitions but unaware of word swaps. The same holds for NegMPNet, which shows performances similar to its base model for word swap, word drop, and repetition perturbations. In contrast, NegBLEURT and NegMPNet clearly outperform their base models with a mean difference score of up to 0.5 for the negation perturbation. These results demonstrate that our models are aware of negation but do not overfit on them and, thus, preserve their performance on tasks aside pure negation detection.

5.2.2 DEMETR benchmark

DEMETR (Karpinska et al., 2022) is a diagnosing benchmark dataset for machine-translation output. It contains reference-candidate pairs with different perturbed versions of the candidates, spanning semantic, syntactic, and morphological errors. These

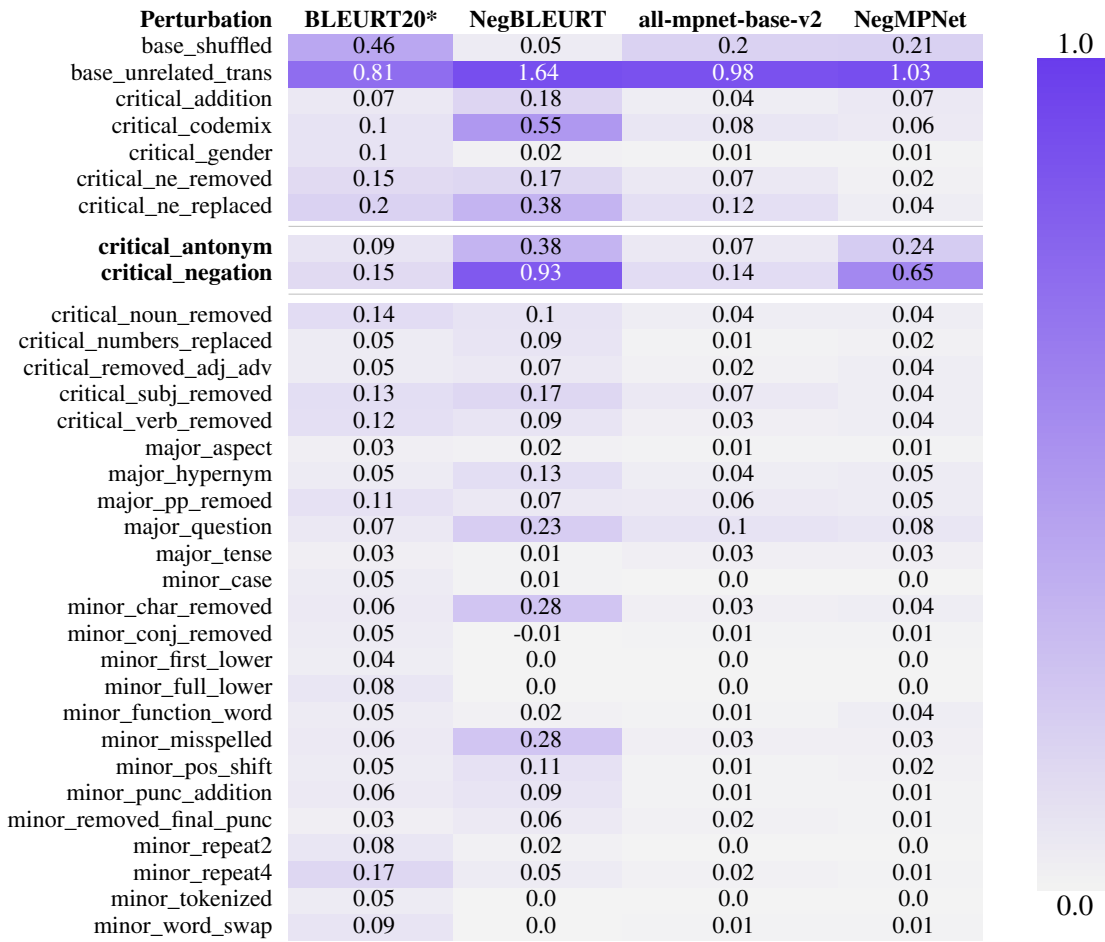| Perturbation	BLEURT20*	NegBLEURT	all-mpnet-base-v2	NegMPNet
base_shuffled	0.46	0.05	0.2	0.21
base_unrelated_trans	0.81	1.64	0.98	1.03
critical_addition	0.07	0.18	0.04	0.07
critical_codemix	0.1	0.55	0.08	0.06
critical_gender	0.1	0.02	0.01	0.01
critical_ne_removed	0.15	0.17	0.07	0.02
critical_ne_replaced	0.2	0.38	0.12	0.04
critical_antonym	0.09	0.38	0.07	0.24
critical_negation	0.15	0.93	0.14	0.65
critical_noun_removed	0.14	0.1	0.04	0.04
critical_numbers_replaced	0.05	0.09	0.01	0.02
critical_removed_adj_adv	0.05	0.07	0.02	0.04
critical_subj_removed	0.13	0.17	0.07	0.04
critical_verb_removed	0.12	0.09	0.03	0.04
major_aspect	0.03	0.02	0.01	0.01
major_hypernym	0.05	0.13	0.04	0.05
major_pp_remoed	0.11	0.07	0.06	0.05
major_question	0.07	0.23	0.1	0.08
major_tense	0.03	0.01	0.03	0.03
minor_case	0.05	0.01	0.0	0.0
minor_char_removed	0.06	0.28	0.03	0.04
minor_conj_removed	0.05	-0.01	0.01	0.01
minor_first_lower	0.04	0.0	0.0	0.0
minor_full_lower	0.08	0.0	0.0	0.0
minor_function_word	0.05	0.02	0.01	0.04
minor_misspelled	0.06	0.28	0.03	0.03
minor_pos_shift	0.05	0.11	0.01	0.02
minor_punc_addition	0.06	0.09	0.01	0.01
minor_removed_final_punc	0.03	0.06	0.02	0.01
minor_repeat2	0.08	0.02	0.0	0.0
minor_repeat4	0.17	0.05	0.02	0.01
minor_tokenized	0.05	0.0	0.0	0.0
minor_word_swap	0.09	0.0	0.01	0.01

Figure 4: Sensitivity scores of different NLG metrics on the DEMETR benchmark. The values represent ratios as introduced in Karpinska et al. (2022). A higher value denotes a higher sensitivity and is marked in a darker color. Both our models clearly have a higher sensitivity towards negations than their base versions.
* copied from the original paper

errors are categorized by their severity, with categories being critical, major, and minor. The authors measured the sensitivity of the metric to a specific perturbation by predicting the metrics' scores for the reference-candidate and reference-perturbed candidate pairs and calculating the weighted difference between the scores. In their original work, BERTScore achieved the best negation (0.21) and antonym (0.15) detection scores. However, these values still need to catch up to detection scores of other perturbations.

Figure 4 shows the sensitivity scores of our metrics compared to their base versions. Sensitivity scores for other metrics such as BERTScore or COMET are presented on the original DEMETR paper by Karpinska et al. (2022). NegBLEURT clearly outperforms all metrics on the critical negation and antonym perturbations while preserving or even improving the detection rate on other perturbations compared to BLEURT20 (Pu et al., 2021).

Both all-mpnet-base-v2 and NegMPNet show no sensitivity for most of the perturbations, indicating that these sentence transformers were not trained for the task of NLG evaluation. Nevertheless, NegMPNet shows a competitive detection rate on negations and even antonyms. An increased sensitivity towards antonyms indicates that our fine-tuning approach yields embeddings that distinguish between negated and affirmative sentences beyond the presence of the word "not". Although our dataset mainly focuses on simple verbal negations, fine-tuning on it teaches the models to capture antonym-related nuances better. This suggests that our models do not simply learn the artefacts in our CANNOT dataset by hard but can distinguish between different types of contradictions.

6 Ablation study

Our CANNOT dataset consists of a diverse collection of datasets, as presented in (section 3).

DEMETR Perturbation	Neg-BLEURT	w/o Not another Negation Benchmark (Truong et al., 2022b)	w/o Automated Fact-Checking of Claims from Wikipedia (Sathe et al., 2020)	w/o GLUE Diagnostic Dataset (Wang et al., 2018)	w/o Sentiment-annotated reviews (Kotzias et al., 2015) with rule-based negations
numbers_replaced	0.09	0.06	0.04	0.11	0.05
gender	0.02	0.03	0.02	0.02	0.03
shuffled	0.05	0.04	0.02	0.06	0.04
adj_adv_removed	0.07	0.06	0.04	0.09	0.09
verb_removed	0.09	0.1	0.06	0.01	0.11
noun_removed	0.1	0.3	0.07	0.21	0.15
subj_removed	0.17	0.09	0.1	0.2	0.09
ne_removed	0.17	0.14	0.1	0.17	0.15
codemix	0.55	0.33	0.31	0.52	0.5
addition	0.18	0.2	0.05	0.18	0.17
antonym	0.38	0.29	0.1	0.41	0.39
negation	0.93	0.74	0.35	1.02	0.82
ne_replaced	0.38	0.26	0.25	0.4	0.4
Number of removed samples	0	281	53.747	399	7.475

Table 3: Ablation study of CANNOT subsets. Each data source in the CANNOT-WMT data was removed individually, and the resulting BLEURT checkpoint evaluated on the DEMETR perturbations marked as critical (Karpinska et al., 2022). Cells with an orange color indicate a decreased performance compared to NegBLEURT, while green indicates an improvement. The darker the color, the larger the difference.

Here, we perform an ablation study to measure the impact of the individual subsets on the model's overall performance. We individually removed each subset from the CANNOT-WMT training and evaluation data, and fine-tuned a new BLEURT checkpoint based on the remaining subsets and the WMT data. The setup for training was the same as for NegBLEURT, which means we fine-tuned the BLEURT-tiny checkpoint for 500 steps using the fine-tuning script from the BLEURT GitHub page. We did not modify the CANNOT-WMT test split, and thus, the number of samples in the subsets deviate from the numbers in section 3.

To compare the impacts of the different datasets, we evaluated the fine-tuned models on all critical perturbations in the DEMETR evaluation benchmark (Karpinska et al., 2022). The model's sensitivities towards specific perturbations are presented in Table 3. The Wiki-Factcheck dataset (Sathe et al., 2020) is by far the biggest subset. As expected, removing it from the training data results in sensitivity drops for nearly all perturbations, especially for antonyms and negations. Removing the Truong et al. (2022b) dataset from the training data results in a substantial sensitivity loss towards the negation perturbation. This is especially remarkable as the dataset is very small, with less than 300 samples. The BLEURT checkpoint fine-tuned without the GLUE subset shows an increased sensitivity towards negations, indicating that the dataset contains some noise introduced by the selection of contradicting samples. These samples may cover contradictions beyond pure negations that decrease NegBLEURT's performance.

7 Conclusion

In this work, we created a sentence negation tool that we made available to the research community as a Python package. In addition, we released CANNOT, a data collection for negation detection that can be used to improve negation awareness of language models. We leveraged this dataset to fine-tune a sentence transformer and an NLG evaluation metric. Both models show a strong negation

detection ability while preserving task-specific performances compared to their base models. Considering the many papers that pointed out the negation weaknesses of model language models, our work is an important step towards negation awareness.

In the future, we will extend our negator to support more advanced negations beyond the verb level and make our negation dataset multilingual so that multilingual NLG evaluation metrics can be improved as well.

8 Limitations

Our rule-based negation system works on a verb level and fails for cases that do not match our defined sentence structure. In addition, there are special cases like the sentence "She's determined" that could use both the verbs "is" and "has". When removing contractions, the negator has to select one of the verbs and may, hence, change the meaning of the input sentence beyond the pure negation. Moreover, in sentences like "I have not yet been there.", the adverb "yet" must be removed or replaced by "already", both of which still need to be added to our tool.

We evaluated our negation-aware models on two evaluation metric benchmark datasets. We would have wished to extend this evaluation to further benchmarks, but unfortunately, the lack of published code or datasets, and insufficient code documentation prevented us from doing so.

While NegMPNet achieves remarkable negation detection scores on the considered metric evaluation benchmarks, we must admit that it fails with most of the other perturbations. The all-mpnet-base-v2 model was initially trained as a sentence transformer and not as an evaluation metric, and the small percentage of WMT data in our dataset is insufficient to train it to be such. Therefore, NegMP-Net can produce negation-sensitive sentence embeddings but needs further work to be applied as an evaluation metric directly.

Ethical Statement

As stated in section 1, trustworthy automatic evaluation metrics are indispensable for selecting and deploying large language models. Metrics that capture negations and reduce the overall score for models that mix up negated and original sentences are, therefore, an important step to increase trust in the metrics themselves, but also in the evaluated models. Moreover, models can be trained to improve

negation sensitivity with metrics that detect negation insensitivity, as well as the CANNOT dataset. Therefore, we do not see ethical concerns with our negation-aware metrics or datasets.

However, our negation tool can add or remove a negation to any input sentence. If applied to sentences from the Internet, such as news articles or Twitter posts, it can easily alter the information provided. The negated and original versions still look very similar, and thus, people might oversee the missing or added negation cues when comparing the provided information with other sources. Consequently, we are aware that our negator may be used in a malicious way to spread misinformation. Nevertheless, negation-aware sentence embeddings and evaluation metrics could again detect such modifications. We believe that the benefits of an open-source tool for researchers, as well as the simplified dataset creation it enables, outweigh the drawbacks of potential misuse.

Supplementary Materials Availability Statement: All material used in this paper is available to the research community. The sentence negation tool is published as a Python package and in a GitHub repository. The dataset and source code for fine-tuning on this data is also open-sourced on GitHub and Hugging Face. The checkpoints of our models are available on the Hugging Face Model Hub. The links to the individual resources are referenced in their respective paper sections.

References

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models

partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks.

Chadi Helwe, Simon Coumes, Chloé Clavel, and Fabian Suchanek. 2022. TINA: Textual inference with negation augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4086–4099, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1301–1312, Online. Association for Computational Linguistics.

Marzena Karpinska, Nishant Raj, Katherine Thai, Yixiao Song, Ankita Gupta, and Mohit Iyyer. 2022. DEMETR: Diagnosing evaluation metrics for translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9540–9561, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Aditya Khandelwal and Suraj Sawant. 2020. NegBERT: A transfer learning approach for negation detection and scope resolution. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5739–5748, Marseille, France. European Language Resources Association.

Philipp Koch, Matthias Aßenmacher, and Christian Heumann. 2022. Pre-trained language models evaluating themselves - a comparative study. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 180–187, Dublin, Ireland. Association for Computational Linguistics.

Dimitrios Kotzias, Misha Denil, Nando de Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page

597–606, New York, NY, USA. Association for Computing Machinery.

Wai Ching Leung, Shira Wein, and Nathan Schneider. 2022. Semantic similarity as a window into vector- and graph-based metrics. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 106–115, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Andrew Moore and Jeremy Barnes. 2021. Multi-task learning of negation and speculation for targeted sentiment classification. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2838–2869, Online. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ananya B. Sai, Tanay Dixit, Dev Yashpal Sheth, Sreyas Mohan, and Mitesh M. Khapra. 2021. Perturbation CheckLists for evaluating NLG evaluation metrics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7219–7234, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated fact-checking of claims from Wikipedia. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6874–6882, Marseille, France. European Language Resources Association.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pretraining for language understanding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Thinh Truong, Timothy Baldwin, Trevor Cohn, and Karin Verspoor. 2022a. Improving negation detection with negation-focused pre-training. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4188–4193, Seattle, United States. Association for Computational Linguistics.

Thinh Hung Truong, Yulia Otmakhova, Timothy Baldwin, Trevor Cohn, Jey Han Lau, and Karin Verspoor. 2022b. Not another negation benchmark: The NaN-NLI test suite for sub-clausal negation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 883–894, Online only. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A MTEB full results

Task	Benchmark	Model	
		all-mpnet-base-v2*	NegMPNet
Classification (12 datasets)	AmazonCounterfactualClassification	65.27	73.96
	AmazonPolarityClassification	67.13	86.1
	AmazonReviewsClassification	31.92	41.85
	Banking77Classification	81.86	84.23
	EmotionClassification	39.72	45.98
	ImdbClassification	70.72	68.4
	MTOPDomainClassification	92.08	93.38
	MTOPIntentClassification	70.21	78.45
	MassiveIntentClassification	69.57	74.38
	MassiveScenarioClassification	76.01	78.12
	ToxicConversationsClassification	60.86	66.15
	TweetSentimentExtractionClassification	55.46	58.99
Retrieval (20 datasets)	ArguAna	46.52	19.51
	CQADupstackAndroidRetrieval	56.49	53.46
	CQADupstackEnglishRetrieval	52.29	49.36
	CQADupstackGamingRetrieval	60.03	52.16
	CQADupstackGisRetrieval	44.27	40.42
	CQADupstackMathematicaRetrieval	34.21	31.8
	CQADupstackPhysicsRetrieval	50.97	44.38
	CQADupstackProgrammersRetrieval	44.17	41.24
	CQADupstackStatsRetrieval	38.15	36.38
	CQADupstackTexRetrieval	33.35	30.7
	CQADupstackUnixRetrieval	45.41	42.18
	CQADupstackWebmastersRetrieval	44.24	42.92
	CQADupstackWordpressRetrieval	35.94	33.39
	DBPedia	32.09	23.08
	FiQA2018	49.96	26.89
	NFCorpus	33.29	27.97
	SCIDOCS	23.76	20.1
	SciFact	65.57	30.83
	TRECCOVID	51.33	58.9
	Touche2020	19.93	16.72
Clustering (11 datasets)	ArxivClusteringP2P	48.38	42.53
	ArxivClusteringS2S	39.72	37.92
	BiorxivClusteringP2P	39.62	33.7
	BiorxivClusteringS2S	35.02	33.45
	MedrxivClusteringP2P	35.58	29.97
	MedrxivClusteringS2S	32.87	31.48
	RedditClustering	54.82	44.31
	RedditClusteringP2P	56.77	45.43
	StackExchangeClustering	53.8	49.4
	StackExchangeClusteringP2P	34.28	30.14
	TwentyNewsgroupsClustering	49.74	44.7
Reranking (3 datasets)	AskUbuntuDupQuestions	65.85	65.11
	SciDocsRR	88.65	87.75

| Task | Benchmark | Model | |
		all-mpnet-base-v2*	NegMPNet
	StackOverflowDupQuestions	51.98	51.87
STS (10 datasets)	BIOSSES	80.43	64.45
	SICK-R	80.59	76.71
	STS12	72.63	71.23
	STS13	83.48	84.62
	STS14	78	79.39
	STS15	85.66	84.7
	STS16	80.03	82.17
	STS17	90.6	90.77
	STS22	67.95	57.63
	STSBenchmark	83.42	84.18
PairClassification (3 datasets)	SprintDuplicateQuestions	90.15	77.69
	TwitterSemEval2015	73.85	75.98
	TwitterURLCorpus	85.11	83.48
Summarization (1 datasets)	SummEval	27.49	29.84

Table 4: Detailed performance on MTEB by task and Benchmark dataset.
* copied form the official leaderboard at `https://huggingface.co/spaces/mteb/leaderboard`, as of 17.05.2023.