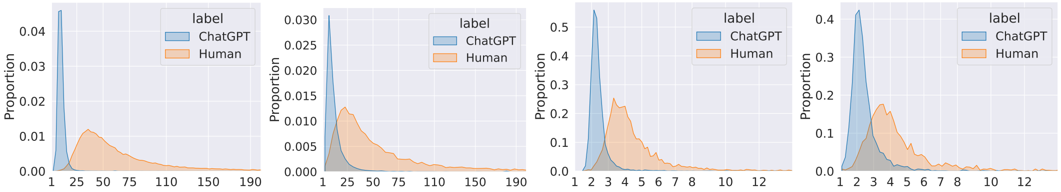

(a) English text ppl (b) English sent ppl (c) Chinese text ppl (d) Chinese sent ppl

Figure 4: PPL distributions on both English and Chinese data, as well as both text and sentence levels.

the LM. A lower PPL indicates that the language model is more confident in its predictions, and is therefore considered to be a better model. The training of LMs is carried out on large-scale text corpora, it can be considered that it has learned some common language patterns and text structures. Therefore, we can use PPL to measure how well a text conforms to common characteristics.

We use the open-source GPT-2 small[8] (Wenzhong-GPT2-110M[9] for Chinese) model to compute the PPL (both text-level and sentence-level[10] PPLs) of the collected texts. The PPL distributions of text written by humans and text generated by ChatGPT are shown in Figure 4.

It is clearly observed that, regardless of whether it is at the text level or the sentence level, the content generated by ChatGPT has relatively lower PPLs compared to the text written by humans. ChatGPT captured common patterns and structures in the text it was trained on, and is very good at reproducing them. As a result, text generated by ChatGPT have relatively concentrated low PPLs.

Humans have the ability to express themselves in a wide variety of ways, depending on the context, audience, and purpose of the text they are writing. This can include using creative or imaginative elements, such as metaphors, similes, and unique word choices, which can make it more difficult for GPT2 to predict. Therefore, human-written texts have more high-PPL values, and show a long-tailed distribution, as demonstrated in Figure 4.

5 ChatGPT Content Detection

AI-generated content (AIGC) is becoming increasingly prevalent on the internet, and it can be difficult to distinguish it from human-generated content, as shown in our human evaluation (sec 3.1). Therefore, AIGC detectors are needed to help identify and flag content that has been created by a machine, to reduce the potential risks to society caused by improper or malicious use of AI models, and to improve the transparency and accountability of the information that is shared online.

In this section, we conduct several empirical experiments to investigate the ChatGPT content detection systems. Detecting AI-generated content is a widely studied topic [19, 27]. Based on these [30, 13, 27], we establish three different types of detection systems, including machine learning-based and deep learning-based methods, and evaluate them on different granularities and data sources. Detailed results and discussions are provided.

5.1 Methods

Detection of machine-generated text has been gaining popularity as text generation models have advanced in recent years[19, 27]. Here, we implement three representative methods from classic machine learning and deep learning, i.e, a logistic regression model trained on the GLTR Test-2[13] features, a deep classifier for single-text detection and a deep classifier for QA detection. The deep classifiers for both single-text and QA are based on RoBERTa [22], a strong pre-trained Transformer [35] model. In fact, algorithms for OOD detection or anomaly detection [17] can also be applied to develop ChatGPT content detectors, which we leave for future work.

[8] https://huggingface.co/gpt2

[9] https://huggingface.co/IDEA-CCNL/Wenzhong-GPT2-110M

[10] For English text, we used NLTK[3] for sentence segmentation (HarvestText for Chinese).

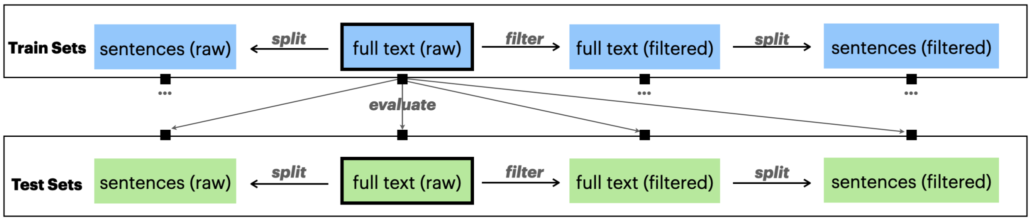

Figure 5: The experiment design for the training and testing of detectors. Different dataset versions are generated through filtering or splitting.

GLTR. [13] studied three tests to compute features of an input text. Their major assumption is that to generate fluent and natural-looking text, most decoding strategies sample high probabilities tokens from the head of the distribution. We select the most powerful Test-2 feature, which is the number of tokens in the Top-10, Top-100, Top-1000, and 1000+ ranks from the LM predicted probability distributions. And then a logistic regression model is trained to finish the classification.

RoBERTa-*sinlge*. A deep classifier based on the pre-trained LM is always a good choice for this kind of text classification problem. It is also investigated in many studies and demo systems [30, 9, 27]. Here we fine-tune the RoBERTa [22] model.

RoBERTa-*QA*. While most content detectors are developed to classify whether a single piece of text is AI-generated, we claim that a detector that supports inputting both a question and an answer can be quite useful, especially for question-answering scenarios. Therefore, we decide to also build a QA version detector. The RoBERTa model supports a text pair input format, where a separating token is used to join a question and its corresponding answer.

5.2 Implementation Details

For the LM used by GLTR, we use gpt2-small [28] for English, and Wenzhong-GPT2-110M released by [36] for Chinese, it is the same with sec. 4.4. For RoBERTa-based deep classifiers, we use `roberta-base`[11] and `hfl/chinese-roberta-wwm-ext`[12] checkpoints for English and Chinese, respectively. All the above models are obtained from huggingface `transformers` [37].

We train the logistic regression model by sklearn [26] on the GLTR Test-2 features from trainset, and search hyper-params following the code of [27]. The RoBERTa-based detectors are trained by the facilities of `transformers`. Specifically, we use the AdamW optimizer, setting batch size to 32 and learning rate to $5e-5$. We finetune models by 1 epoch for English, and 2 epochs for Chinese.

5.3 Experiment Design

The HC3 dataset consists of questions and their corresponding human/ChatGPT answers. We extracted all the `<question, answer>` pairs, and assigned label `0` to pairs with human answers and label `1` to pairs with ChatGPT answers.

Simply using the original answers from humans and ChatGPT to train a binary classifier is the most straightforward way. However, there might be some issues by doing so:

- First, based on the observations in Section 3, both human answers and ChatGPT answers may contain some obvious indicating words that may influence the effectiveness of models;
- Second, users may want to detect whether a single sentence is generated by ChatGPT, instead of the full text. This can be quite difficult for a classifier that is only trained on full texts;
- Third, taking the corresponding question of the answer into account may help the detector to make a more accurate judgment, compared with only considering the answer itself. This

[11] `https://huggingface.co/roberta-base`
[12] `https://huggingface.co/hfl/chinese-roberta-wwm-ext`

can be widely applied to many QA platforms (like Quora, Stack Overflow, and Zhihu) to find out which answer below a certain question is generated by AI.

Therefore, we design different groups of experiments to study these key questions:
• How will the indicating words influence the detector?
• Is it more challenging for the ChatGPT detectors to detect sentence-level content? Is it harder to train a sentence-level classifier?
• Can the corresponding question help detectors detect the origin of the answer more accurately?

Figure 5 shows how we generate different types of training and testing sets. Specifically, we use the collected raw corpus to construct the first train-test sets (the "full text (raw)" in the figure), which we call the ***raw-full*** version. Then we filter away the indicated words in the text to obtain the ***filtered-full*** version. By splitting the full text into sentences, we obtain the ***raw-sent*** version and the ***filtered-sent*** version. We also combine the full text and the sentences into a mixed version, namely the ***raw-mix*** and ***filtered-mix*** version. Overall, we have six different versions of training and testing sets. Evaluating a model's performance on version B's testing set which is trained on version A's training set can be seen as an out-of-distribution (OOD) generalization evaluation, which is more challenging since it requires the model to be robust when facing sample style changes.

5.4 Results

Following the above experiment design, we conduct comprehensive empirical studies on all kinds of derived corpus. Table 4 shows the test F1 scores.

		English							**Chinese**						
Test →		*raw*			*filtered*			Avg.	*raw*			*filtered*			Avg.
		full	sent	mix	full	sent	mix		full	sent	mix	full	sent	mix	
Train ↓		**RoBERTa**													
raw	full	99.82	81.89	84.67	99.72	81.00	84.07	88.53	98.79	83.64	86.32	98.57	82.77	85.85	89.32
	sent	99.40	98.43	98.56	99.24	98.47	98.59	**98.78**	97.76	95.75	96.11	97.68	95.31	95.77	**96.40**
	mix	99.44	98.31	98.47	99.32	98.37	98.51	98.74	97.70	95.68	96.04	97.65	95.27	95.73	96.35
filtered	full	99.82	87.17	89.05	99.79	86.60	88.67	91.85	98.25	91.04	92.30	98.14	91.15	92.48	93.89
	sent	96.97	97.22	97.19	99.09	98.43	98.53	97.91	96.60	92.81	93.47	97.94	95.86	96.26	95.49
	mix	96.28	96.43	96.41	99.45	98.37	98.53	97.58	97.43	94.09	94.68	97.66	95.61	96.01	95.91
Train ↓		**GLTR Test-2**													
raw	full	98.26	71.58	76.15	98.22	70.19	75.23	81.61	89.61	44.02	53.72	85.89	43.58	53.62	61.74
	sent	86.26	88.18	87.96	87.72	88.23	88.19	87.76	84.49	71.79	74.01	84.06	70.29	72.90	76.26
	mix	95.97	86.45	87.81	96.13	86.24	87.73	90.06	86.45	70.85	73.59	84.94	69.14	72.14	76.19
filtered	full	98.31	70.91	75.65	98.30	69.48	74.72	81.23	89.46	58.69	64.52	86.51	55.45	62.18	69.47
	sent	84.00	88.25	87.71	85.68	88.35	87.99	87.00	84.56	71.85	74.07	84.22	70.59	73.18	76.41
	mix	95.36	86.73	87.97	95.60	86.56	87.92	90.02	86.30	71.00	73.70	84.98	69.45	72.40	76.31

Table 4: F1 scores (%) of different models on each testset, average of each language are reported.

5.4.1 Which detector(s) is more useful? ML-based or DL-based? and Why?

According to Table 4, we can derive following conclusions:

Firstly, **the robustness of RoBERTa-based-detector is better than GLTR**. The F1-scores of RoBERTa decrease slightly (1.5-2% in English datasets and 2-3% in Chinese datasets) when sentences are split by comparing the leading diagonal elements in *raw→raw* and *filtered→filtered*. In contrast, the GLTR reduces significantly by over 10% in English datasets, and above 15% in Chinese datasets. Above all, the RoBERTa-based-detector is more robust with anti-interference character. In contrast, the GLTR reduces significantly by over 10% in English datasets, above 15% in Chinese datasets. Above all, the RoBERTa-based-detector is more robust with anti-interference character.

Secondly, **RoBERTa-based-detector is not affected by indicating words.** The F1-scores of RoBERTa only slightly decreased by 0.03% in English *full* dataset, and 0.65% in Chinese *full* dataset, as seen in the minus of relevant leading diagonal elements in *raw→raw* versus *filtered→filtered*. On the contrary, evaluations based on GLTR decrease by up to 3.1% on Chinese datasets, though tiny rise on English datasets, indicating that GLTR is sensitive to indicating words, easily influenced by the patterns of ChatGPT.