**[SeqXGPT: Sentence-Level AI-Generated Text Detection](https://arxiv.org/abs/2310.08903)**

**Conclusion**

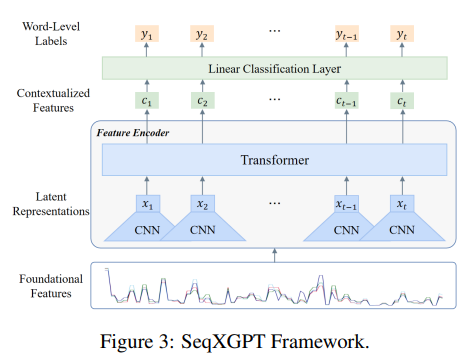
Using SeqXGPT that depends on convolution and self attention networks, to detect AIGT on sentence level not document level using sequence labeling, and log probability that outputs a wave pattern for texts which is later classified using the same methods in speech recognition that deals with waves.

Log probability, Z scores, perplexity\* are used as main features based on either word wise or token wise features depending on test.

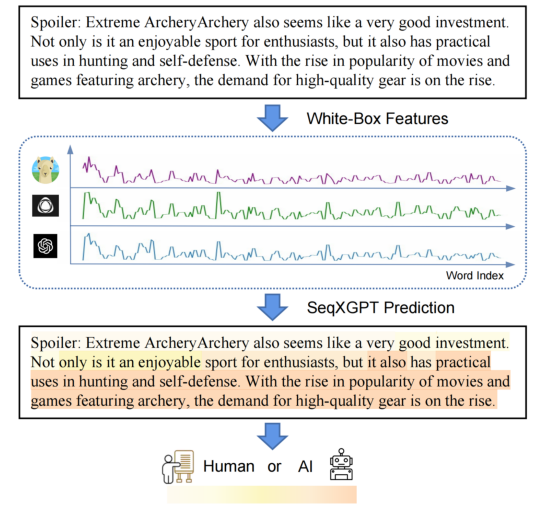
Data collected are from different models ex: GPT 2, GPT 2 neo, GPT J, LLMA, GPT 3, Human, the different models are also used in detecting the origin of the generated text.

The comparison was between the base model(Seq XGPT), Sniffer, Sent Robert, and Seq- Roberta.

The SeqXGPT achieved high results in both document and sentence level detection and good generalization in the Out Of Distribution (OOD) dataset



**Abstract**

* Most AIGT Detection focuses on Document level detection, in this paper we try to work more on Sentence level detection.
* Document level detection is less effective on single sentences.
* Sequence XGPT a model that uses Log probability lists features. These features are composed like waves in speech processing.
* This method suprasses tests in both document level and sentence level.

**Introduction**

* Perplexity is the main feature, using word wise log probability, as foundational feature.
* SeqXGPT uses CNN and self attention networks to process the waves.
* Select most frequent word wise category.
* SeqXGPT has excellent generalization.
* This method focuses more on intrinsic features, like log probability, token rank, predictive entropy.

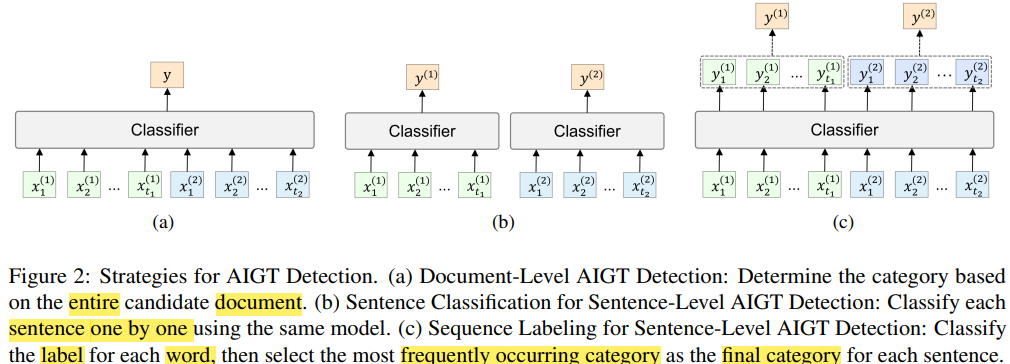
**Detection formulas**

1. Particular model binary detection: Detect text produced by specific model ex: GPT zero, detect GPT
2. Mixed model binary detection: identify Ai text without stating exact model
3. Mixed model multiclass: Detect text and identify the model that generated it.

**Sentence level detection**

Document generated by several models, is hard to get detected and causes a lot of false negatives and false positives error. The sentence level checks each sentence in whole document giving higher accuracy.

Uses 2 methods, Sentence classification which treats each sentence as input and classifies it, and Sequence labelling which takes all document as input and labels every word and the label the sentence with the most appeared label for words in it (AI- human)



**Approaches**

* Uses lop probability that differentiate through perplexity\*
* Compute perplexity for each sentence and draw histogram to show distribution.
* Manually select threshold to determine whether sentence is human or AI generated.
* Add multiple perturbations (noise or changes), and compute Z-score and draw histogram.

**Brief on other models**

**SeqXGPT**: Uses sequence labeling, which consists of 1) perplexity extraction and anlignment, 2)Feature Encoder, 3)linear classification layer

**Roberta:** Based on transformer encoder, that can classify and label sequences. Used 2 versions one for labeling (Seq Roberta) and detection (Sent-roberta)

**Sniffer:** Powerful that can detect the model of generated text, used base model but changed data to sentence data instead of documents.

\*\*Perplexity is a metric used to measure how well probability model predicts a sample, [for more info](https://chat.openai.com/share/e15f89c7-fdac-48bf-9bda-5b68541a60bf)