Building a better AI detector using retrieval methods

Dev Bhatia
Computer Systems Lab Dr. Yilmaz
Period: 3
Date: 2/25/2025
Project Presentation 3

Problem

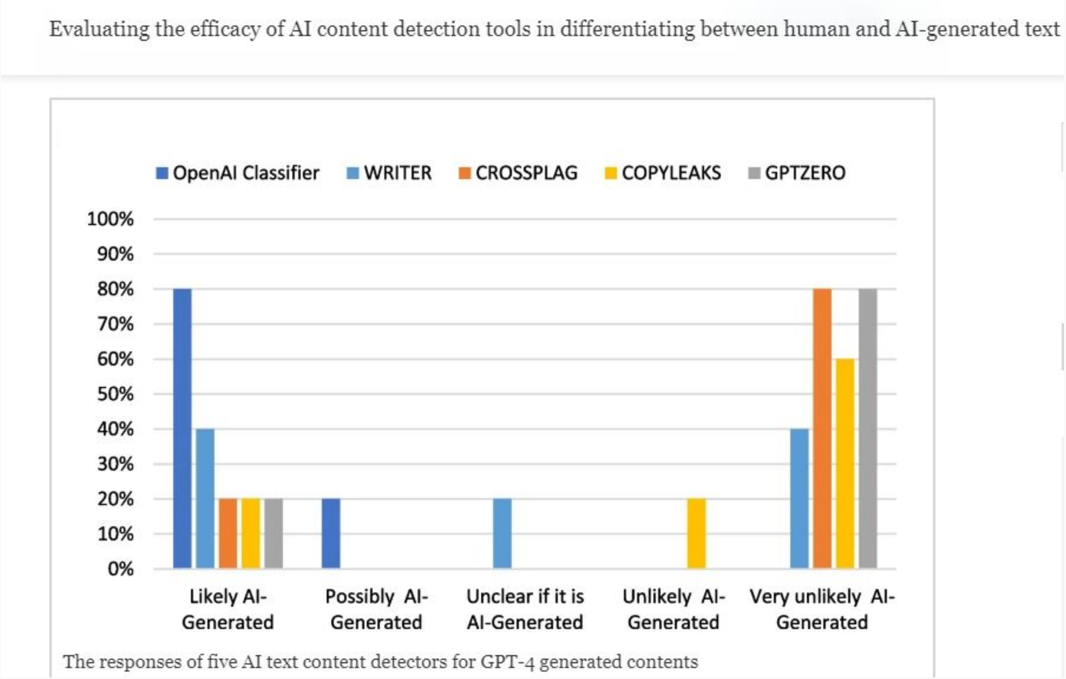
- AI used very prominently
- People tend to paraphrase AI content
- Difficulty telling original work from AI generated text
- Current AI detectors don’t work well
- Leads to lack of knowledge for students
- AI gives untrustworthy or false information
- Need better detection mechanism

Background

- Many AI detectors out there
- Accuracy varies significantly
- Unreliable for teacher use
- Possibility of false positives
- Not close to 100% accuracy

Other Solutions

- Most prominent AI detectors:
  - GPTZero [6]
  - Copyleaks [7]
  - CrossPlag [8]
- Detectors do not perform well against GPT 4.0 and other AIs
- Many false negatives and uncertain classifications



Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text

The responses of five AI text content detectors for GPT-4 generated contents

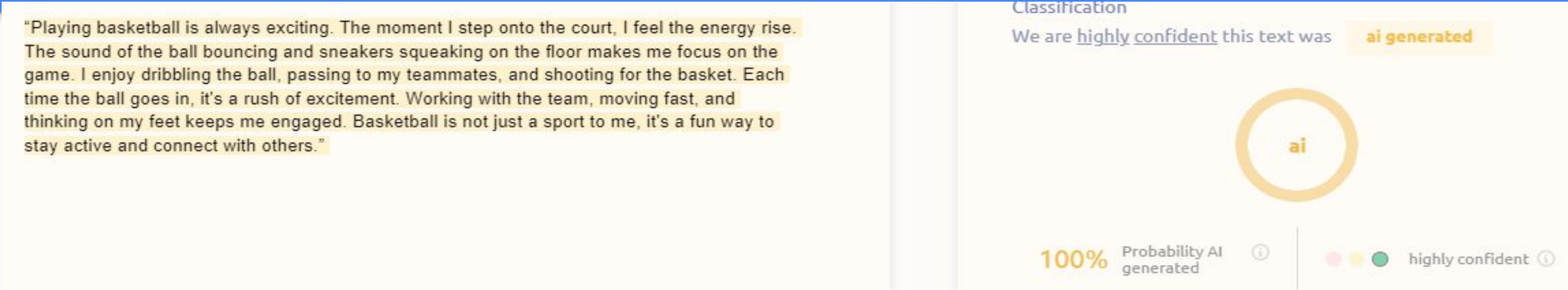
Other Solutions Continued

- How these AI detectors work
  - GPTZero:
    - Amount of predictability in the text
    - Looks at the variance in the sentences, AI generated text typically has less variety
  - Copyleaks
    - Scans the document against different sources to see if it matches anything
    - Sentence by sentence detection against human writing
  - CrossPlag
    - Text analysis using Natural Language Processing
    - Using a dataset created by human and AI content
- Pros: Detect well on older GPT generations
- Cons: Fail on GPT 4.0 and paraphrased AI generations, inconsistencies

Demo of current solutions

- GPTZero:
- CopyLeaks: (doesn't seem to work without subscription)
- CrossPlag:
- Sample AI generated text: “Playing basketball is always exciting. The moment I step onto the court, I feel the energy rise. The sound of the ball bouncing and sneakers squeaking on the floor makes me focus on the game. I enjoy dribbling the ball, passing to my teammates, and shooting for the basket. Each time the ball goes in, it’s a rush of excitement. Working with the team, moving fast, and thinking on my feet keeps me engaged. Basketball is not just a sport to me, it’s a fun way to stay active and connect with others.”

AI Detectors with paraphrasing continued



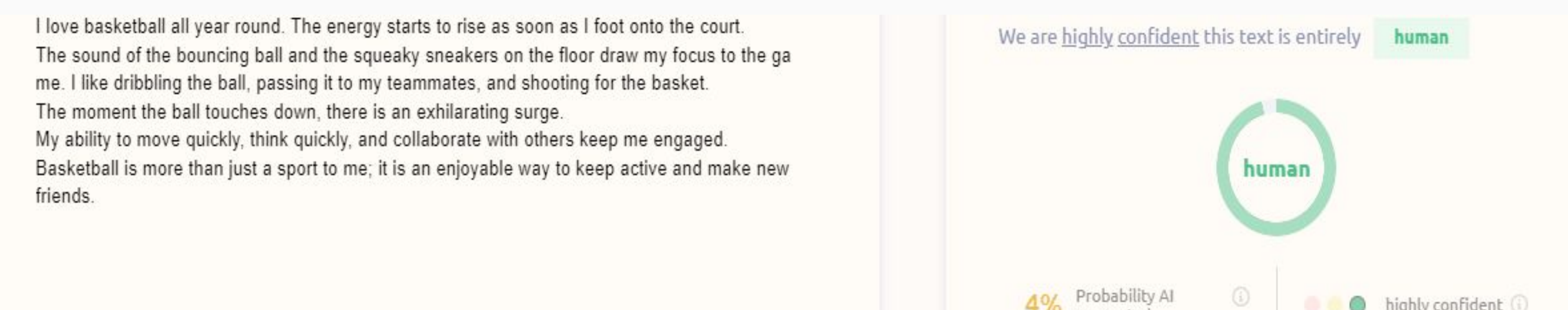
"Playing basketball is always exciting. The moment I step onto the court, I feel the energy rise. The sound of the ball bouncing and sneakers squeaking on the floor makes me focus on the game. I enjoy dribbling the ball, passing to my teammates, and shooting for the basket. Each time the ball goes in, it's a rush of excitement. Working with the team, moving fast, and thinking on my feet keeps me engaged. Basketball is not just a sport to me, it's a fun way to stay active and connect with others."

Classification

We are highly confident this text was ai generated

100% Probability AI generated

● ● ● highly confident



I love basketball all year round. The energy starts to rise as soon as I foot onto the court.
The sound of the bouncing ball and the squeaky sneakers on the floor draw my focus to the game. I like dribbling the ball, passing it to my teammates, and shooting for the basket.
The moment the ball touches down, there is an exhilarating surge.
My ability to move quickly, think quickly, and collaborate with others keep me engaged.
Basketball is more than just a sport to me; it is an enjoyable way to keep active and make new friends.

We are highly confident this text is entirely human

4% Probability AI

● ● ● highly confident

Why Is Mine Better?

- Model that is more accurate on GPT 4.0
    - Current detectors struggle on this
- Model that is better with paraphrased GPT 4.0
    - GPT 4.0 content that people paraphrase but the meaning is the same
- Model that gives more consistency in detecting GPT 4.0 content
- Model that works better with DeepSeek, Gemini, o3, Grok

Novelty

- Using retrieval methods (not used on common AI detectors)
  - Using a database of AI generations to tell whether text is a paraphrasing of AI generated text
    - Use cosine similarity scores
    - Find any matches to previous generations
  - Accuracy will remain high even as the amount of paraphrasing goes up
- Use on “mixing attacks”
  - Texts that has both human written elements and AI generated text
- Higher accuracy than current AI detecting algorithms have

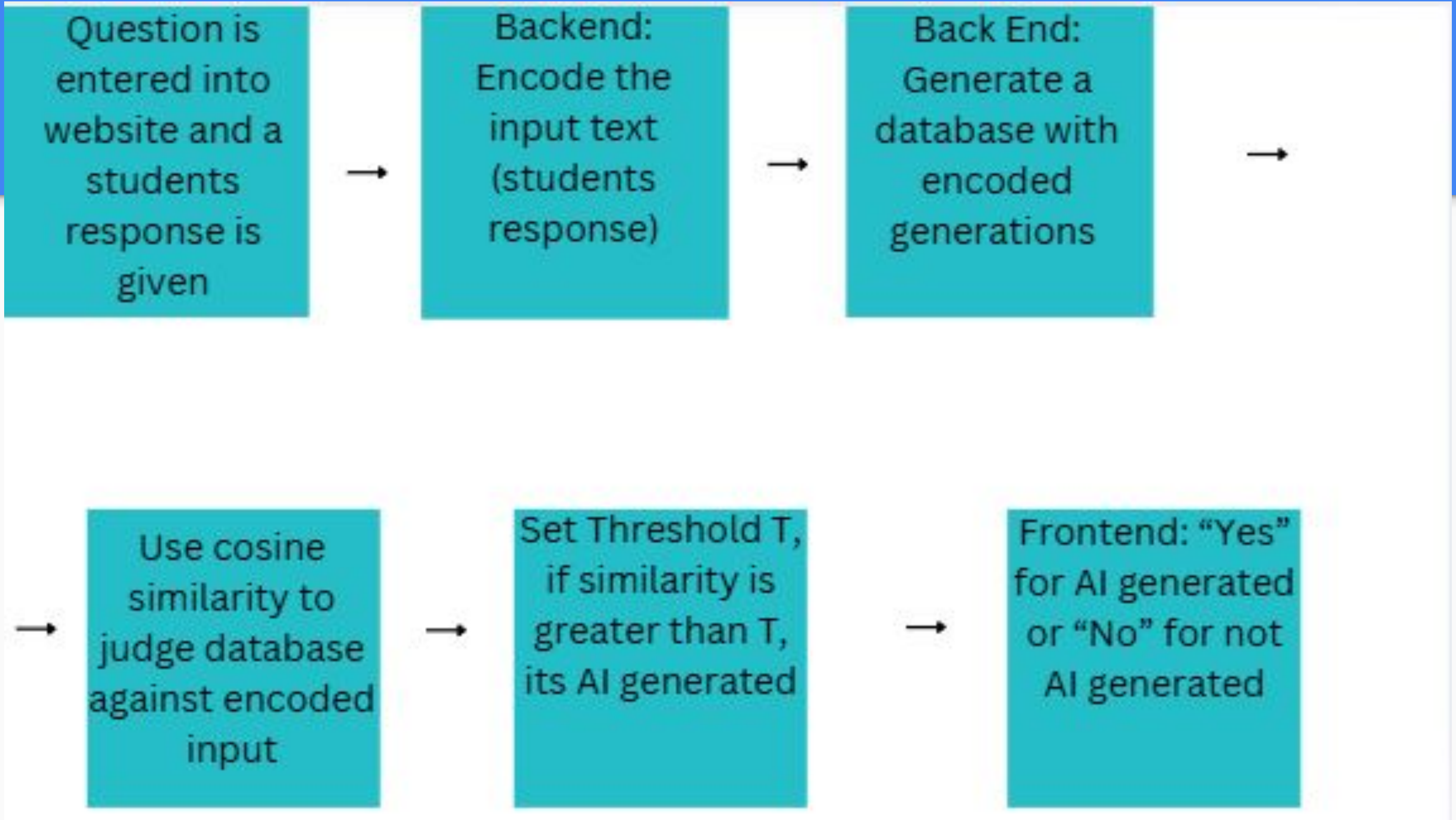
Impact

- Used in school by teachers
  - A more reliable way to check AI content
- Trustworthy source of AI detection
- Maintains integrity
- Creates a fair learning environment

Method

- Store Doc2Vec encoded sequences of text that were generated by AI in database in a web server
- Feed in input text
- Use a Doc2Vec encoder to encode input text
- Check to see if input text matches database text (cosine similarity)
- Set the threshold (T), if the text score is higher than T, then it is judged as AI Generated
- Output whether the text is detected as AI generated or not

Method: System Architecture



Question is entered into website and a students response is given →

Backend: Encode the input text (students response) →

Back End: Generate a database with encoded generations →

→ Use cosine similarity to judge database against encoded input →

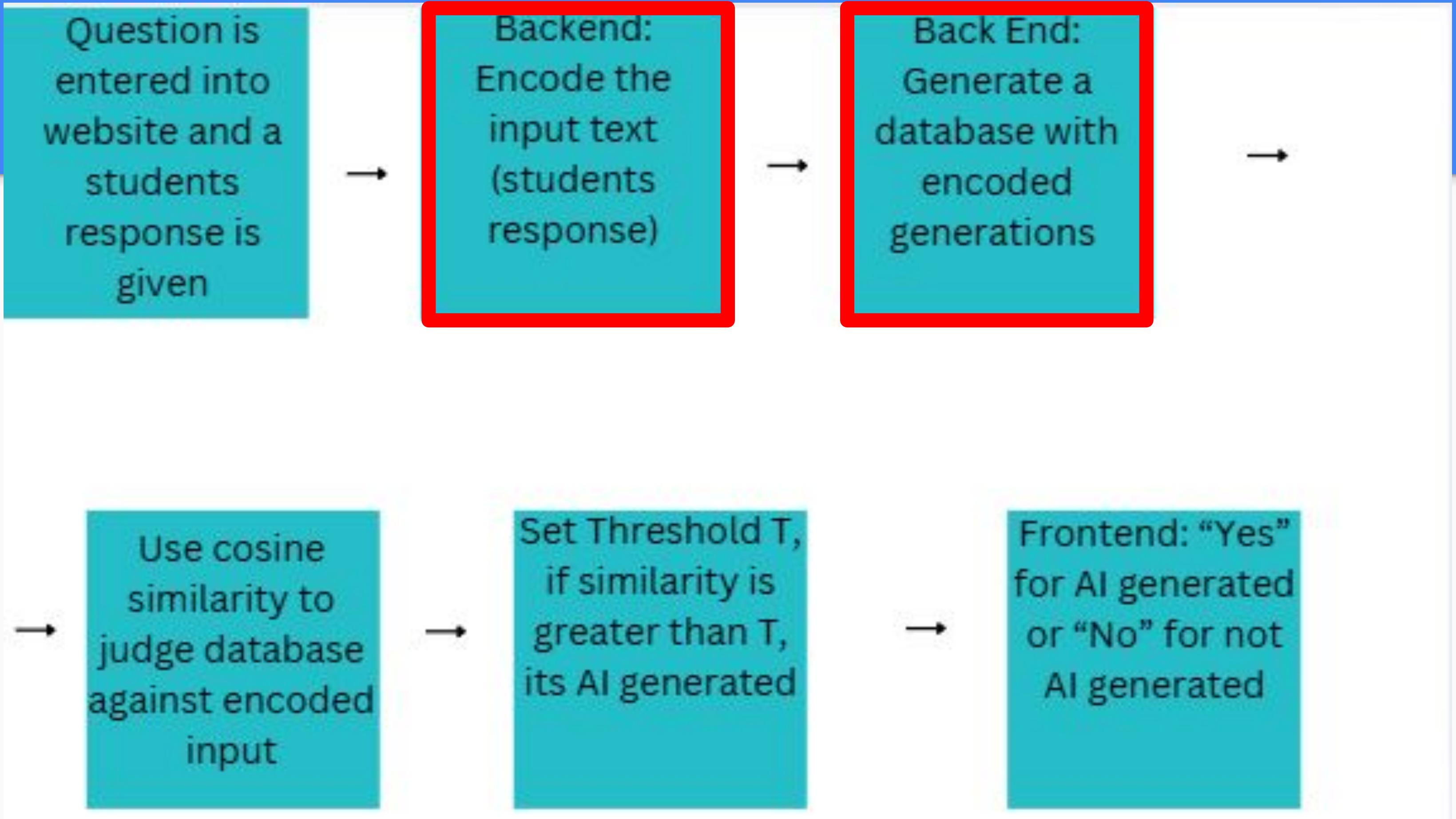
Set Threshold T, if similarity is greater than T, its AI generated →

Frontend: "Yes" for AI generated or "No" for not AI generated

Process Overview

- Example:
  - Question (Entered by teacher): Describe the sky
  - Input Text (Sample response): "The sky is clear and blue"
  - Database of generations:
    - “The weather is sunny and bright” Vector: [0.45,0.25,0.65,0.1,0.35,0.3]
    - “Today is a clear blue day” Vector: [0.4,0.2,0.6,0.1,0.3,0.2]
    - “The sky is bright and blue” Vector: [0.51,0.31,0.69,0.12,0.41,0.22]
  - Input Text Vector Representation: [0.5,0.3,0.7,0.1,0.4,0.2]
  - Cosine Similarity "The sky is clear and blue" and "The sky is bright and blue" = 0.999
  - Threshold = 0.95
  - 0.999 > 0.95, therefore AI generated

Process Overview



Question is entered into website and a students response is given

Backend: Encode the input text (students response)

Back End: Generate a database with encoded generations

Use cosine similarity to judge database against encoded input

Set Threshold T, if similarity is greater than T, its AI generated

Frontend: “Yes” for AI generated or “No” for not AI generated

Encoding - Doc2Vec

- Generates a vector representation of a document
- Purpose: Captures the context of the overall document. Similar documents have similar vectors.
- Uses a 2-layer neural network.

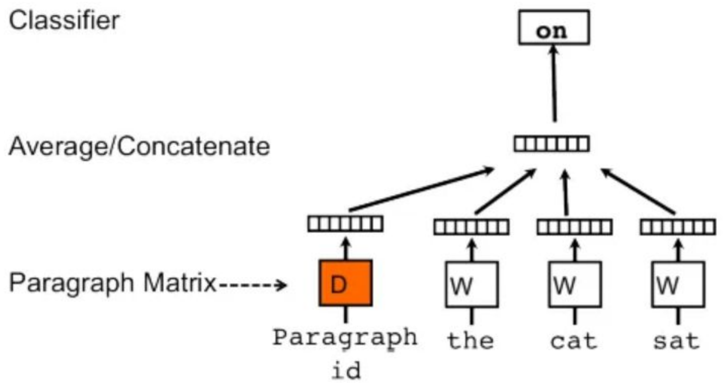


fig 3: PV-DM model

Doc2Vec Continued

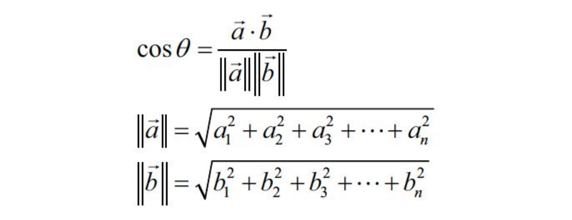
- Distributed Memory version of Paragraph Vector (PV-DM)
  - Input Layer: context words and a document ID
    - Originally the words and Doc ID is a unique vector (initialized randomly)
  - Hidden Layer: Combined Input (average). Activation Function (ReLU)
  - Use softmax function to calculate probability of each word
  - Output Layer: Target word
  - Keep updating the vectors until it gets toward desired result (minimize the loss until you get closer to the target probability distribution)
  - Sliding window of words (a sentence) moves around the document, and the context words and the Document ID are used to predict the target word

Database

- Created using generations around a prompt or topic
- Teacher can put a prompt in and generate AI responses
- Can generate hundreds of pieces of text for the database
- Take the LLM output and encode it as a vector
- Store the vector in the database (Doc2Vec)

Cosine Similarity

- Gives a value in the range 0 to 1
- Close to 1 means the generations are similar
- Close to 0 means the generations are not similar
- Example:
- Document 1: [1, 1, 1, 1, 1, 0] let’s refer to this as A
- Document 2: [1, 1, 1, 1, 0, 1] let’s refer to this as B
- Cosine Similarity = (4) / (2.2360679775*2.2360679775) = 0.80 (80 percent similarity between the two sentences)

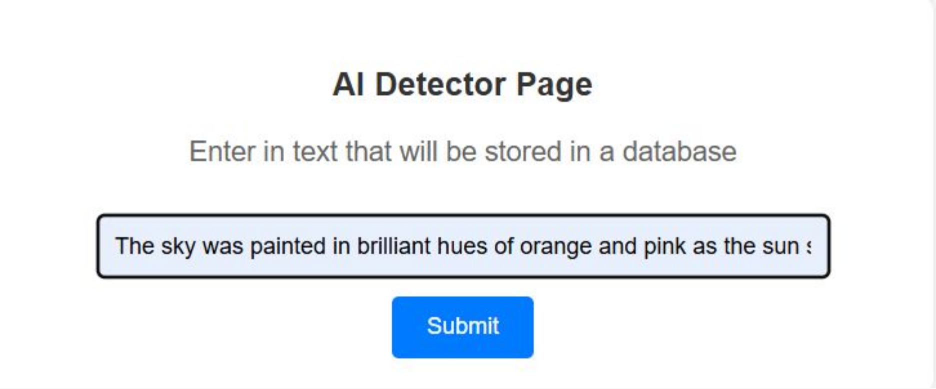


$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

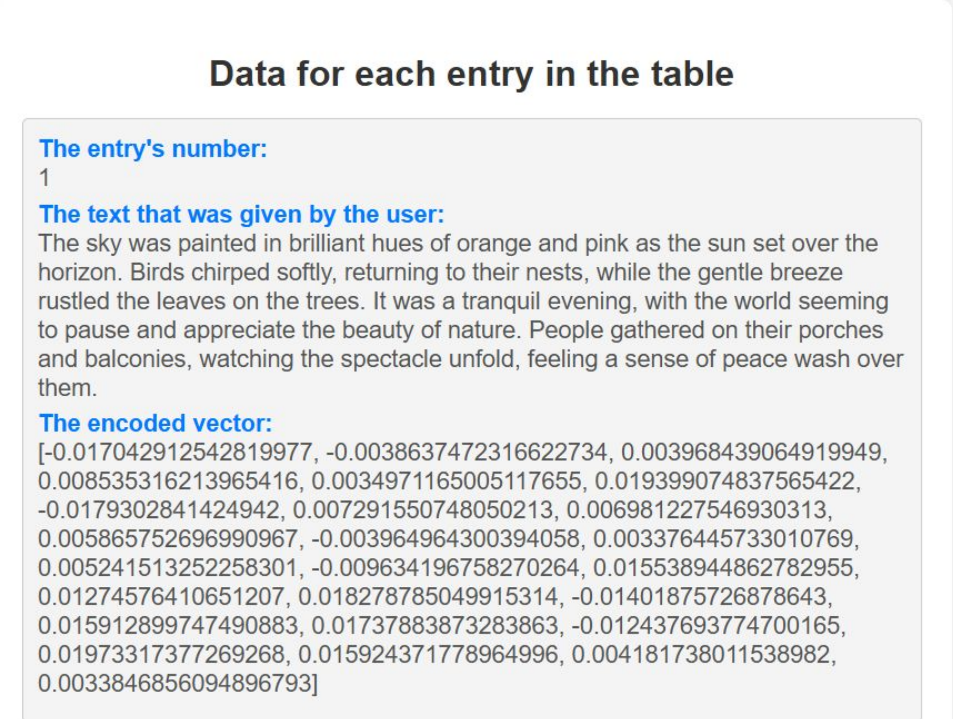
Demo



AI Detector Page

Enter in text that will be stored in a database

The sky was painted in brilliant hues of orange and pink as the sun



Data for each entry in the table

The entry's number:
1

The text that was given by the user:
The sky was painted in brilliant hues of orange and pink as the sun set over the horizon. Birds chirped softly, returning to their nests, while the gentle breeze rustled the leaves on the trees. It was a tranquil evening, with the world seeming to pause and appreciate the beauty of nature. People gathered on their porches and balconies, watching the spectacle unfold, feeling a sense of peace wash over them.

The encoded vector:
```
[-0.017042912542819977, -0.0038637472316622734, 0.003968439064919949,
0.008535316213965416, 0.0034971165005117655, 0.019399074837565422,
-0.0179302841424942, 0.007291550748050213, 0.006981227546930313,
0.005865752696990967, -0.003964964300394058, 0.003376445733010769,
0.005241513252258301, -0.009634196758270264, 0.015538944862782955,
0.01274576410651207, 0.018278785049915314, -0.01401875726878643,
0.015912899747490883, 0.01737883873283863, -0.012437693774700165,
0.01973317377269268, 0.015924371778964996, 0.004181738011538982,
0.0033846856094896793]
```

Results

- I hope to measure 85% accuracy or better on detecting text generated by AI
- Model takes input text
  - Could be paraphrased
  - Could be mixed in with human content,
  - Model tells whether it was AI generated or not
- Expect better performance than current AI detectors

Limitations

- Text not generated by students
- Text with shorter number of words
- Not using watermarking and statistical outlier methods
- Text written in other languages

Future Work

- Improvements in text generated for other things besides school
- Use retrieval method along with other methods (watermarking, statistical outlier) to improve shorter text identification
- Using retrieval methods with AI generations in other languages

Conclusion

- I solved the problem of AI Detection using Retrieval Methods

References

[1] A. Singh, "A Comparison Study on AI Language Detector," *2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2023, pp. 0489-0493, doi: 10.1109/CCWC57344.2023.10099219.

[2] D. Dukić, D. Keča and D. Stipić, "Are You Human? Detecting Bots on Twitter Using BERT," *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, Sydney, NSW, Australia, 2020, pp. 631-636, doi: 10.1109/DSAA49011.2020.00089.

[3] Elkhatat, A.M., Elsaid, K. & Almeer, S. Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text. *Int J Educ Integr* 19, 17 (2023). https://doi.org/10.1007/s40979-023-00140-5

[4] Krishna, Kalpesh, et al. "Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense." *Advances in Neural Information Processing Systems* 36 (2024).

[5] Perkins, Mike, et al. "Simple Techniques to Bypass GenAI Text Detectors: Implications for Inclusive Education: Revista De Universidad y Sociedad Del Conocimiento." *International Journal of Educational Technology in Higher Education*, vol. 21, no. 1, 2024, pp. 53. *ProQuest*, https://www.proquest.com/scholarly-journals/simple-techniques-bypass-genai-text-detectors/docview/3101842024/se-2, doi:https://doi.org/10.1186/s41239-024-00487-w.

[6]E. Tian, “GPTZero,” *gptzero.me*, 2022. https://gptzero.me/

[7] “Copyleaks: AI & Machine Learning Powered Plagiarism Checker,” *copyleaks.com*. https://copyleaks.com/ (accessed Jan. 23, 2025).

[8]“Crossplag,” *app.crossplag.com*. https://app.crossplag.com/individual/detector (accessed Jan. 23, 2025).

Q&A

THANKS!