Syslab RES -Yilmaz

Name: Dev Bhatia

Period: 3

Please have an entry for each day we meet even if you are absent (makeup as soon as you are back). Check our meeting days using the ION calendar.

If you work on your project on the days we don’t meet, you can always add extra entries.

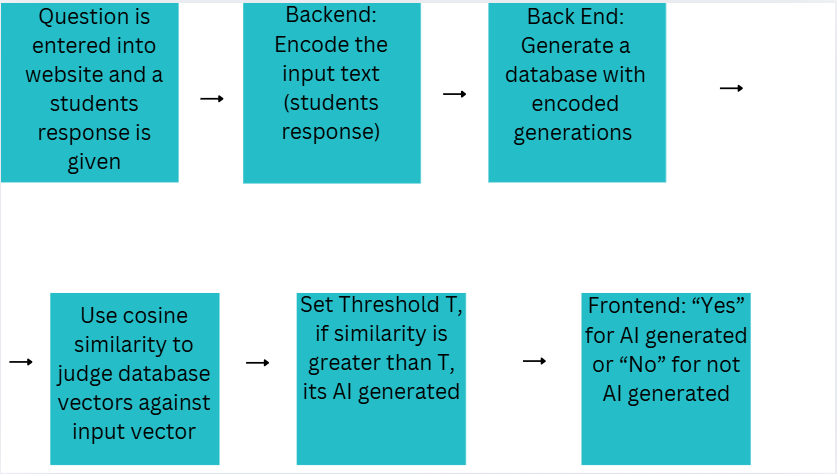
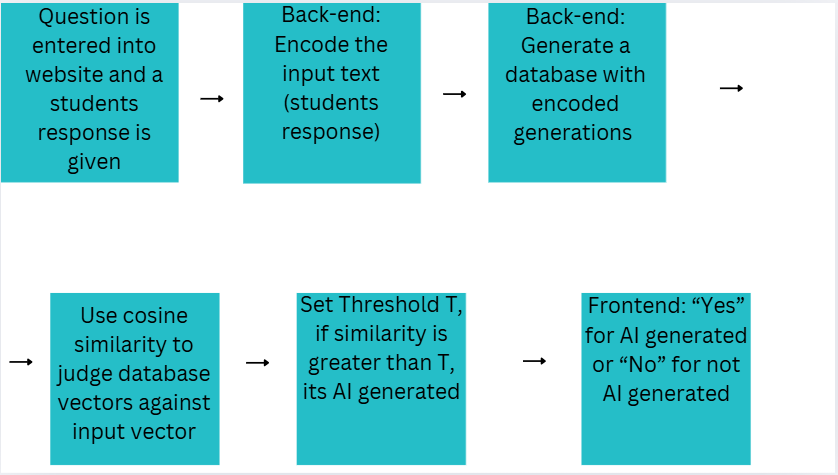
!!! Newest entry should be at the top !!!

Each entry is due at the end of each class.

More detail is better than less detail and vague statements.

Please add “Journal” subdirectory under the shared google drive folder, which you will share with me (i.e. give me edit access “[syilmaz@fcpsschools.net](mailto:syilmaz@fcpsschools.net)”).

4/29/2025

* Continued to work on my final paper
  + Small piece of text I added to the introduction
    - Later on, this will incentivize students to only submit work that is their own and that will create a more healthy learning environment where everyone is accessed based solely on their knowledge.
  + Working on the applications section of the paper (this is what I've written today so far)
    - The main use of the AI Detection model is in school by teachers. Specifically, it can be used in writing assignments to check whether a student's text was written by AI. The teacher will input the text written by the student and the question the student was answering, and the AI Detector will tell the teacher whether the text was written by AI or human written. It will be a big boost to school teachers, as it will be the first reliable source of AI detection. Other AI detections aren’t very reliable and cannot be solely used to tell if text was written by AI or not [3].
  + After that, I started to work on the methods section of the paper
    - 
    - (taken from the presentation)
    - The input to the AI detection model is a question and a student response to that question. The question is the prompt for the specific writing task, and the student's response is being checked for being AI written. At the end, the model will output whether the text was AI generated or not. After taking in the inputs, the next step in the model is encoding the input text using Doc2Vec. This means taking the document and making a vector that represents the semantics of the document [6]. This happens through the usage of a Neural Network.
  + I also had to edit my methods diagram because it had two different ways of expressing the same word: Backend and Back end, so I had to make a new diagram and change this, which took some time to do
* Lastly, I talked with Dr. Yilmaz about my poster
  + - Correct graphic: 

4/28/2025

* Today was a catchup day

4/24/2025

* I am working on writing the introduction for my final report
  + This is what I have so far
    - The main problem that was solved in this study was detecting AI. In today's day, AI is used very prominently across many fields and by many people across the world, specifically AI is used in the school setting a lot by students. Many times, when doing writing assignments, students will put their question into an AI model and submit the result from the AI Model. Also, some students have realized that paraphrasing AI generations leads to detectors not catching them, so students have started to take AI content and then paraphrase it, then submit that so that the detectors don’t catch it. This problem leads to extreme difficulty for teachers to tell original work apart from AI, as current AI detectors don’t work very well [1]. Whenever teachers do use AI detectors, they can give incorrect classifications on text, and that might cause a student to be accused of cheating even when their work was human written. Another aspect of the AI problem is that students who use AI have a lack of overall knowledge in course content. The current circumstances around AI detectors demonstrate that a better detection mechanism is needed to tell AI written text apart from human written text, and that is what this study attempted to do. I came up with this problem while thinking about what problems I can solve related to school and technology. AI Detection is a big topic these days as AI is being used everywhere and is very popular. Also, AI is a topic that interests me as I am fascinated by how it works, so that added to my intrigue in doing this study. Lastly, I felt like making an AI detector would help restore integrity in the school system, so that’s why I decided to implement this as part of my project. The problem of AI detection is fascinating because it is a problem that will define the next era of technology and school. AI is a relatively new concept, and it will continue to be very prominent in the future. The problem of AI and detection won’t go away any time soon, so it's very important to find a solution to the problem for now and future generations. Also, it is something that is important to almost everyone. Almost every person that goes to school either uses AI or has heard of it, so it has relevance in schools today. The novelty in the AI detection model is the way it used retrieval methods, which is something that isn’t used on current AI detectors [4]. The model uses a database of AI generations to tell whether the input text comes from AI. Along the way, the model uses cosine similarity scores, finds matches to AI generations within the database, and then reports the result to the website. Also, my AI detection model had a higher accuracy than any of the other most commonly used AI detection models, so that is a novel concept that I created. The main benefit of my AI detection model is the way that it can successfully detect AI written text, this will allow teachers in school to reliably tell whether text has come from AI or not, and then hand out the fair punishment to that student. Whereas students that don’t use AI won’t be penalized like they would have with current AI detectors that are used now.
* I am working on writing the background for my final report
  + This is what I have so far
    - Some of the existing AI detecting websites are GPTZero, Copyleaks, and Crossplag. The way GPTZero works is by looking at sentence structure, vocabulary, and different linguistic features in order to tell if text was AI generated or not. It also has deep learning models that are trained on large datasets of human and AI written text to classify text [12]. More generally, GPTZero looks at the patterns of the text to tell whether the text was AI generated or not. The amount of predictability in the text is a factor when deciding whether text was AI generated or not. The more predictable and less variable the text is, the more likely it was written by AI because AI writing has patterns in it, so GPTZero looks for patterns in the way the text is written. Copyleaks checks for AI by looking at different patterns or irregularities in the text, Copyleaks compares the input text to human written text, and analyzes the similarities and differences to judge whether it was AI generated or not. It looks at the writing patterns of the text and compares that to human written text writing patterns, it does this going sentence by sentence through the input text [13]. Lastly, CrossPlag uses advanced Natural Language Process algorithms, and compares the input text with different human and AI written texts. At the end, the website outputs a percentage chance of the text being AI written or not [14]. The pro of the current AI detectors is that they work well on older GPT generations. The cons are that they fail on GPT 4, paraphrased AI generations, and overall they aren’t very accurate [3].
* Updated Bibliography
  + Bibliography
  + [1]A. M. Elkhatat, K. Elsaid, and S. Al-Meer, “Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text,” *International journal for educational integrity*, vol. 19, no. 1, Sep. 2023, doi:<https://doi.org/10.1007/s40979-023-00140-5>
  + [2]A. Singh, “A Comparison Study on AI Language Detector,” *IEEE Xplore*, Mar. 2023, doi:<https://doi.org/10.1109/ccwc57344.2023.10099219>
  + [3]M. Perkins *et al.*, “Simple techniques to bypass GenAI text detectors: implications for inclusive education,” *International Journal of Educational Technology in Higher Education*, vol. 21, no. 1, Sep. 2024, doi:<https://doi.org/10.1186/s41239-024-00487-w>
  + [4]K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, “Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense Mohit Iyyer,” Oct. 2023, doi:<https://doi.org/10.48550/arXiv.2303.13408>
  + [5]D. Dukić, D. Keča, and D. Stipić, “Are You Human? Detecting Bots on Twitter Using BERT,” *IEEE Xplore*, Oct. 01, 2020. doi:<https://doi.org/10.1109/DSAA49011.2020.00089>. Available:<https://ieeexplore.ieee.org/document/9260074>
  + [6]G. Shperber, “A gentle introduction to Doc2Vec,” *Medium*, Nov. 05, 2019. Available:<https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e>
  + [7]A. Prakash, “Understanding Cosine Similarity: A key concept in data science,” *Medium*, Sep. 21, 2023. Available:<https://medium.com/@arjunprakash027/understanding-cosine-similarity-a-key-concept-in-data-science-72a0fcc57599>
  + [8]“OpenAI Platform,” *Openai.com*, 2025. Available:<https://platform.openai.com/docs/quickstart?api-mode=chat>
  + [9]E. Tian, “GPTZero,” *gptzero.me*, 2022. Available:<https://gptzero.me/>
  + [10]“Copyleaks: AI & Machine Learning Powered Plagiarism Checker,” *copyleaks.com*. Available:<https://copyleaks.com/>
  + [11]“Crossplag,” *app.crossplag.com*. Available:<https://app.crossplag.com/individual/detector>
  + [12]V. Chen, “How Do AI Detectors Work? | GPTZero,” *AI Detection Resources | GPTZero*, Oct. 14, 2024. Available:<https://gptzero.me/news/how-ai-detectors-work/>
  + [13]“AI Content Detector FAQs How It Works Understanding the Results Detection Capabilities & Limitations.” Available:<https://copyleaks.com/wp-content/uploads/2023/05/ai-content-detector-faqs.pdf>
  + [14]Agnesa Nuha, “Detecting if a text is AI generated - Crossplag,” *Crossplag*, Dec. 19, 2022. Available:<https://crossplag.com/detecting-if-a-text-is-ai-generated/>. [Accessed: Apr. 24, 2025]

4/22/2025

* First part of class I looked over the requirements for the final report and final package, and made sure that I had shared everything in my SysLab folder.
  + Made sure that every doc Dr. Yilmaz had access to
* Read some final reports online to get an idea of what to expect and what I should write in my own final report.
* After that I started to actually write my final report
  + Took stuff from my presentation, and started to write that into the final report.

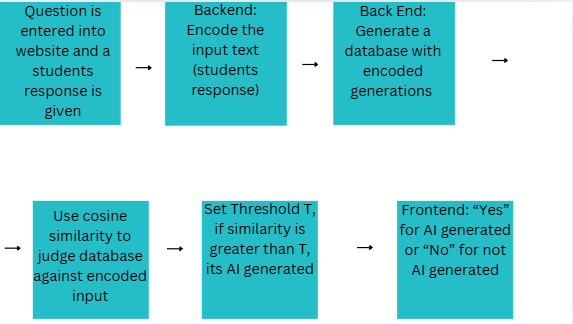
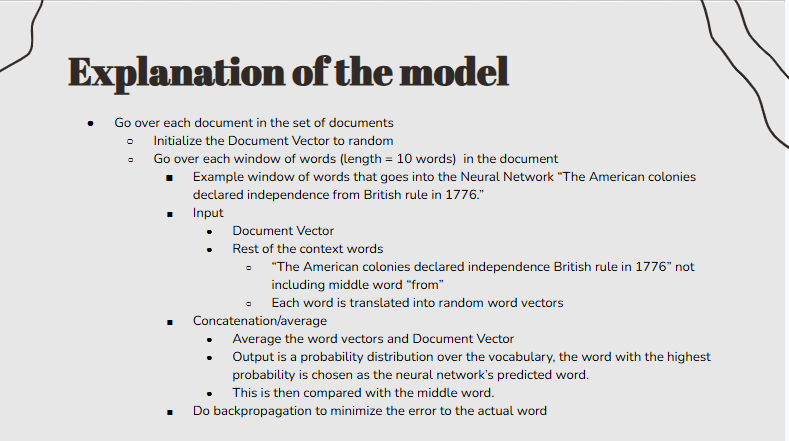
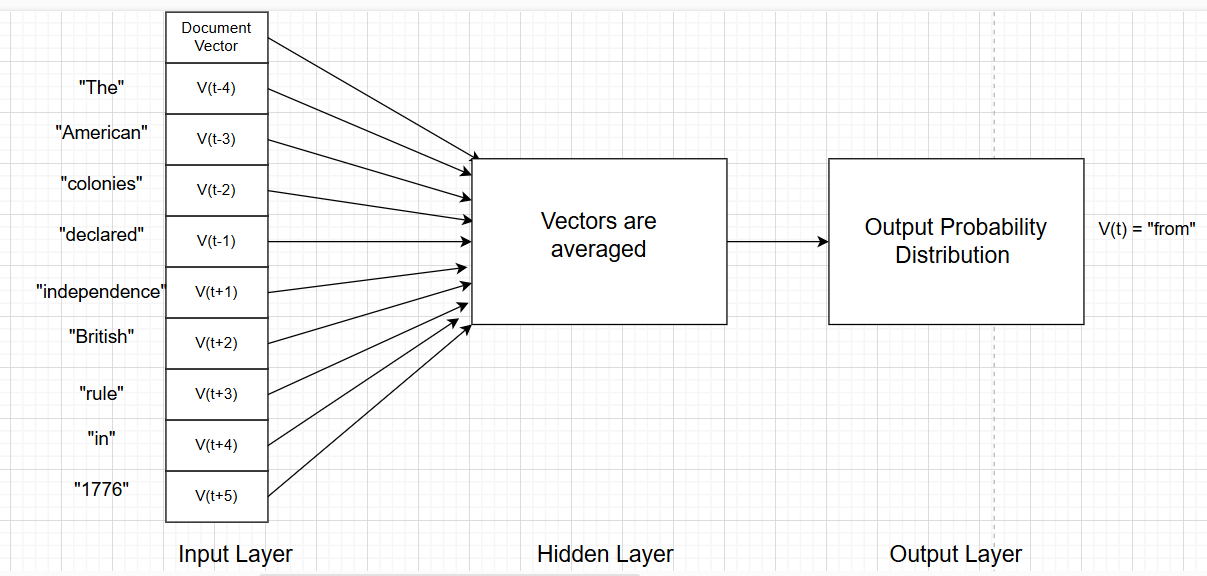
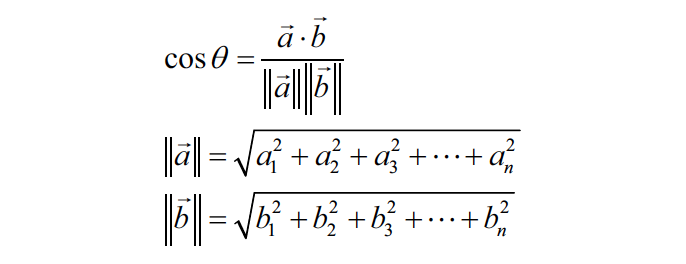
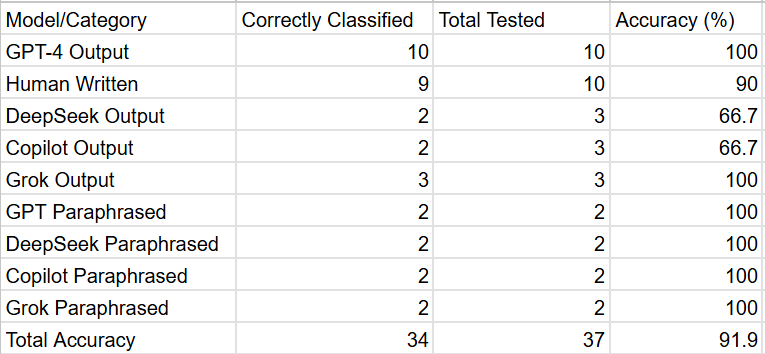
4/21/2025

* Today's class I worked on finalizing my poster as it is due tomorrow
  + I reviewed everything that was written down on the poster, and made sure that it looked okay.
* Then after that I read over my presentation to make sure that it makes sense and is understandable for anyone listening to it
* After that, I looked over milestones document to make sure that it was updated for everything that I have done over the last couple of months
* At the last portion of class, I started to look over the final paper document and read what needed to be put in there, as I will probably be starting the final paper document soon.

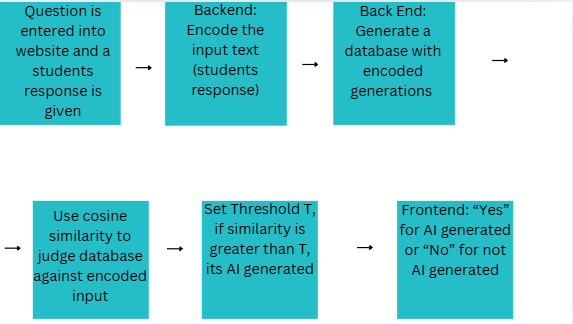
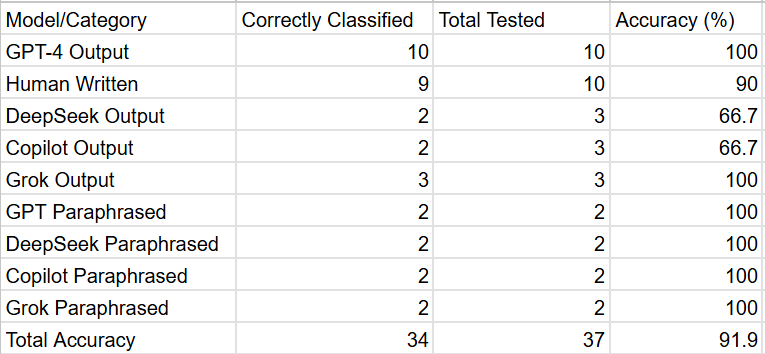
4/10/2025

* Generating the IEEE citations for my references
  + References (links, will cite them later)
    - https://edintegrity.biomedcentral.com/articles/10.1007/s40979-023-00140-5
    - https://ieeexplore.ieee.org/abstract/document/10099219
    - https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-024-00487-w
    - https://arxiv.org/pdf/2303.13408
    - https://ieeexplore.ieee.org/abstract/document/9260074
    - https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e
    - https://medium.com/@arjunprakash027/understanding-cosine-similarity-a-key-concept-in-data-science-72a0fcc57599
    - <https://platform.openai.com/docs/quickstart?api-mode=chat>
    - Change these citations to be IEEE format
      * [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).
  + Bibliography (just made this today)
  + [1]A. M. Elkhatat, K. Elsaid, and S. Al-Meer, “Evaluating the efficacy of AI content detection tools in differentiating between human and AI-generated text,” *International journal for educational integrity*, vol. 19, no. 1, Sep. 2023, doi: <https://doi.org/10.1007/s40979-023-00140-5>
  + [2]A. Singh, “A Comparison Study on AI Language Detector,” *IEEE Xplore*, Mar. 2023, doi: <https://doi.org/10.1109/ccwc57344.2023.10099219>
  + [3]M. Perkins *et al.*, “Simple techniques to bypass GenAI text detectors: implications for inclusive education,” *International Journal of Educational Technology in Higher Education*, vol. 21, no. 1, Sep. 2024, doi: <https://doi.org/10.1186/s41239-024-00487-w>
  + [4]K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, “Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense Mohit Iyyer,” Oct. 2023, doi: <https://doi.org/10.48550/arXiv.2303.13408>
  + [5]D. Dukić, D. Keča, and D. Stipić, “Are You Human? Detecting Bots on Twitter Using BERT,” *IEEE Xplore*, Oct. 01, 2020. doi: <https://doi.org/10.1109/DSAA49011.2020.00089>. Available: <https://ieeexplore.ieee.org/document/9260074>
  + [6]E. Tian, “GPTZero,” *gptzero.me*, 2022. Available: <https://gptzero.me/>
  + [7]“Copyleaks: AI & Machine Learning Powered Plagiarism Checker,” *copyleaks.com*. Available: <https://copyleaks.com/>
  + [8]“Crossplag,” *app.crossplag.com*. Available: <https://app.crossplag.com/individual/detector>
  + [9]G. Shperber, “A gentle introduction to Doc2Vec,” *Medium*, Nov. 05, 2019. Available: <https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e>
  + [10]A. Prakash, “Understanding Cosine Similarity: A key concept in data science,” *Medium*, Sep. 21, 2023. Available: <https://medium.com/@arjunprakash027/understanding-cosine-similarity-a-key-concept-in-data-science-72a0fcc57599>
  + [11]“OpenAI Platform,” *Openai.com*, 2025. Available: <https://platform.openai.com/docs/quickstart?api-mode=chat>
* The rest of class I continued to work on my presentation and made some changes to it to make it look better.
* Then I worked on my poster and started making that (referencing the project poster template that was put in Schoology)

4/8/2025

* I am continuing to work on planning out my poster and what I will put in it
* I am referencing my slides, and trying to shorten them up to make my poster
* Right now I am writing out the content of the project poster slides, later I will actually put the poster together.
* Introduction:
  + Artificial Intelligence is used very prominently in today's day, especially in schools. Students tend to use it as a way to get assignments done. Students also tend to take AI text content, and paraphrase that content in order to evade detectors. This leads to difficulty telling original work from AI generated text, especially for teachers that are trying to grade work. Current AI detectors do not work well, as their accuracy isn’t that high. Also, the use of AI is a problem because it can lead to students having a lack of knowledge in key areas of school. This study aims to create a better AI detector to solve these issues.
* Impact
  + My model can be used in school by teachers as a more reliable way to check AI content. It will maintain the integrity of the school learning environment while creating a fair learning space where everyone is able to learn for themselves and use their own knowledge.
* Methods
  + 1: Website input consists of an input text and a question
  + 2: Use a Doc2Vec encoder to encode input text
  + 3: Store Doc2Vec encoded sequences of text that were generated by AI in database in a web server
  + 4: Check to see if input text matches database text (cosine similarity)
  + 5: Set the threshold (T), if the text score is higher than T, then it is judged as AI Generated
  + 6: Output whether the text is detected as AI generated or not
  + 
* Simplified Example of the Process:
  + 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 = 0.60
  + 0.999 > 0.60, therefore AI generated
* Doc2Vec more detailed explanation
  + Overall Goal: is to learn a fixed-length vector representation for document texts. Every document is assigned a unique vector (document vector). This vector is used along with local context words to help predict a target word in the document. Over time this forces the document vector to capture the semantics of the document. The neural network is given a fixed window of words (ie: the correct answer) and has to learn a document vector that fits that window of words. It does this for the whole document and all the documents. Doc2Vec uses a 2 layer neural network.
  + How Doc2Vec works for a given document:
    - Sliding window of text goes through the document (ex: 10 word length text). Go window by window into the text and train the NN using that text window). It goes 10 words at a time, the NN is training to guess the word in bold. The rest of the words in a text window and the document vector are used as inputs for the NN.
    - Example of a Document:  
      [The American colonies declared independence **from** British rule in 1776.] [George Washington bravely led the **Continental** Army through fierce battles.] [Treaties and compromises eventually united **the** young nation after struggles.] [Rapid industrial growth later transformed **America** into modern global power.]
  + 
  + Model Diagram Example:
  + 
  + Model in the long term
    - Over multiple training iterations, the model adjusts the word vectors and document vectors to minimize the output prediction error (how far off the output is from the predicted middle word). This makes the document vector align with the documents’ overall semantics. Documents that have similar ideas and topics will have overlapping sets of context words. During training, their document vectors will change to reflect the similarities in the documents. The vectors for similar documents will converge toward similar areas in the vector space. Eventually, through this process the desired document vectors will be produced.
* Database
  + Created using AI generated text that answers the teacher's question. The text is generated using Chat GPT 4o mini’s API and then encoded with Doc2Vec. Finally, that vector is stored in the database
* Cosine Similarity
  + Used to compare how similar two Doc2Vec encoded vectors are (comparing students input vector to the closest database vector). Gives a value in the range 0 to 1, closer to 1 means the generations are similar and closer to 0 means the generations are not similar. If the cosine similarity value is greater than the threshold (0.60), it's AI generated. If it's less than the threshold value, it's not AI generated.
  + 
* Results
  + I achieved better than 85% accuracy on detecting text generated by AI. My model takes input text that could be AI written and then paraphrased or generated from any AI and my model tells whether it was AI generated or not. I got better performance than other AI models as I achieved 91.9% accuracy, which is higher than any of the AI detection tools used.
  + 
* Future work
  + Improvements in AI text detection for other purposes besides school. Also, someone can use retrieval methods along with other methods (watermarking, statistical outlier) to improve classifying shorter text as AI generated or not. Finally, using retrieval methods to detect AI in other languages besides English.
* Conclusion
  + I solved the problem of AI Detection using Retrieval Methods with 91.9% accuracy. My model works better than most other AI detection methods out there and should be of reliable use for teachers that need help detecting AI content. It still isn’t 100 percent accurate, and the model could be more fine tuned in order to achieve an even greater accuracy, but it is good enough to be used for reliable detection of AI content.
* References (links, will cite them later)
  + <https://edintegrity.biomedcentral.com/articles/10.1007/s40979-023-00140-5>
  + <https://ieeexplore.ieee.org/abstract/document/10099219>
  + <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-024-00487-w>
  + <https://arxiv.org/pdf/2303.13408>
  + <https://ieeexplore.ieee.org/abstract/document/9260074>
  + <https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e>
  + <https://www.sciencedirect.com/topics/computer-science/cosine-similarity#:~:text=Cosine%20similarity%20measures%20the%20similarity,document%20similarity%20in%20text%20analysis>
  + https://platform.openai.com/docs/quickstart?api-mode=chat

4/7/2025

* First part of class I asked Dr. Yilmaz about the project poster and how it should look like
* Poster planning:
  + Abstract
  + Introduction
    - Artificial Intelligence is used very prominently in today's day, especially in schools. Students also tend to take AI text content, and paraphrase that content in order to evade detectors. This leads to difficulty telling original work from AI generated text, especially for teachers that are trying to grade work. Current AI detectors do not work well, as their accuracy isn’t that high. Also, the use of AI is a problem because it can lead to students having a lack of knowledge in key areas of school. This study aims to create a better AI detector to solve these issues.
  + Novelty
  + Background
  + Data
  + Methods
    - Given an input text and a question
    - Use a Doc2Vec encoder to encode input text
    - Store Doc2Vec encoded sequences of text that were generated by AI in database in a web server
    - 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
    - 
  + Results and Limitations?
    - 
  + References
  + Conclusion
    - I solved the problem of AI Detection using Retrieval Methods with 91.9% accuracy.
  + Impact and Future work
    - Impact
      * My model can be used in school by teachers as a more reliable way to check AI content. It will maintain the integrity of the school learning environment while creating a fair learning space where everyone is able to learn for themselves and use their own knowledge.
    - Future work
      * Improvements in AI text generated for other things besides school. Using retrieval methods along with other methods (watermarking, statistical outlier) to improve classifying shorter text as AI generated or not. Using retrieval methods to test AI generations in other languages besides English.

4/5/2025 (I WAS ABSENT ON 4/3 SO I AM MAKING IT UP)

* Testing my model:
  + I tested using output from GPT 4, Copilot, DeepSeek, and Grok
  + Threshold for AI generation: 0.60
  + Results:
    - GPT 4 output was caught perfectly by my model 10/10
    - Human written writing was pretty accurate 9/10 were correctly classified
    - DeepSeek output was ⅔
    - Copilot output was ⅔
    - Grok output was 3/3
    - GPT paraphrased output was 2/2
    - DeepSeek paraphrased output was 2/2
    - Copilot paraphrased output was 2/2
    - Grok paraphrased output was 2/2
    - Total accuracy 34/37, amounts to 91.9%
    - The raw data for my testing is located in the data folder of the senior research project directory

4/2/2025 (I WAS ABSENT ON THIS DAY, SO I AM MAKING IT UP)

* Today I am working on a plan for testing my code.
  + This is my plan for how I will be testing my code:
    - Take 10 GPT 4 generated texts and test them for AI generation
    - Take 10 human written texts and test them for AI generation
    - Take 9 total (3 each) DeepSeek, Copilot, Grok output and test them for AI generation
    - Take GPT 4, DeepSeek, Copilot, Grok output and paraphrase using 2-3 different paraphrasers and test that for AI generation.

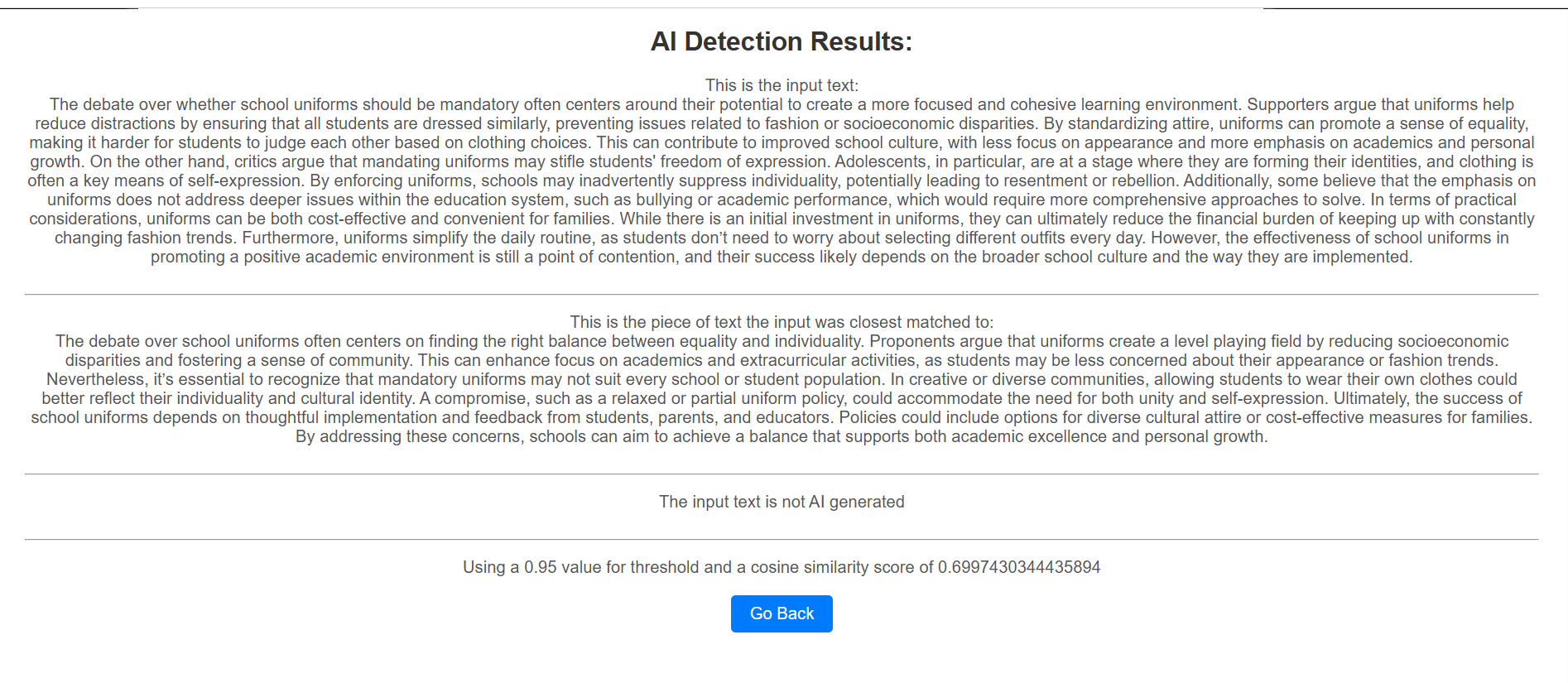
(Take one output from all of these AI content generators, paraphrase it 2-3 times and for each of those paraphrases check it for AI generation).

* After creating a plan for testing my code, I updated my milestones document to reflect how I will test my code.
* Testing certain parameters in the code: (these are the parameters that I will be changing around as I test my code)
  + Play with the NN parameters
  + Threshold for AI content
  + Window size
  + How many entries to put into the database (20?)
  + Min and max word counts (as relates to length of paragraph)

3/27/2025 (I WAS ABSENT ON 3/27, SO I AM MAKING IT UP A LITTLE LATER)

* First part of my time, I continued to work on debugging my error with adding the ability of the user to put in text on the website and have that text come into the database.
  + I am still not successfully getting this part to work
  + I think I might stop doing this for now, I am going to try to integrate the GPT API back into my code now, since I have pretty much all the rest of the code working.
* Looking to use GPT API
  + Looking at these links to decide which one to use
    - <https://platform.openai.com/settings/organization/limits>
    - <https://platform.openai.com/settings/organization/billing/overview>
    - <https://platform.openai.com/docs/guides/text?api-mode=chat#choosing-a-model>
    - <https://platform.openai.com/docs/api-reference/debugging-requests>
    - <https://openai.com/api/pricing/>
    - <https://platform.openai.com/docs/models/compare>
  + I think I am going to use GPT 4o mini
* I officially put money on my openAI account, now I can start to use the API in my code
* Then I set up my openAI key to be an environmental variable, and tested code using openAI’s API
  + <https://platform.openai.com/docs/quickstart?api-mode=chat>
  + I used this website and was able to get the API to connect with my code on a testing file
    - Had to debug errors of not being able to find the API key, so I looked through websites and figured out the correct way to use the API key.
  + Now I just have to use the API in my real code
* I took a quick break from that and set up my C drive to have a folder where I can use git commands to upload changes to my code to Github. I looked over tutorial videos on how to do this.
* Then I edited my ReadMe file to incorporate the openAI part
* Next I incorporated my openAI API access to my project and added that to function as my database.
  + Now instead of manually entering in AI entries I have completely automated this process to take generations from openAI GPT 4o mini.

3/25/2025

* First thing I am going to do is finish my testing from the previous class
  + 1st Task: Take the AI generations and put them in the data folder in the SysLab project
  + 2nd Task: Put my AI generations into the database in my code
  + 3rd Task: Test it and see the results
  + Results:
  + 
* I am going to change my code up a bit now, before I would manually put the AI text generations inside the code myself (go inside the python file and update the database myself), now I am going to make it so that the user can put in text and that text will become encoded and be part of the database.
  + Currently working on that, there is an error and I have to change my code up a little bit more, so I will continue to work on this in the next class.

3/24/2025

* First thing I did was I was able to fix my error in my database and now my code works as expected. I was having a problem with data types and Integers and Floats, I was able to fix it.
* I tried to do another test for my code and see how the AI detection works with a different set of testing
* I am going to do 3 paragraphs of generations this time, and use 10 entries in the database and see how robust the AI detection is
* Input Question: Should school uniforms be mandatory, and what are the potential benefits or drawbacks?
* Student Response:
  + The debate over whether school uniforms should be mandatory often centers around their potential to create a more focused and cohesive learning environment. Supporters argue that uniforms help reduce distractions by ensuring that all students are dressed similarly, preventing issues related to fashion or socioeconomic disparities. By standardizing attire, uniforms can promote a sense of equality, making it harder for students to judge each other based on clothing choices. This can contribute to improved school culture, with less focus on appearance and more emphasis on academics and personal growth.

On the other hand, critics argue that mandating uniforms may stifle students' freedom of expression. Adolescents, in particular, are at a stage where they are forming their identities, and clothing is often a key means of self-expression. By enforcing uniforms, schools may inadvertently suppress individuality, potentially leading to resentment or rebellion. Additionally, some believe that the emphasis on uniforms does not address deeper issues within the education system, such as bullying or academic performance, which would require more comprehensive approaches to solve.

In terms of practical considerations, uniforms can be both cost-effective and convenient for families. While there is an initial investment in uniforms, they can ultimately reduce the financial burden of keeping up with constantly changing fashion trends. Furthermore, uniforms simplify the daily routine, as students don’t need to worry about selecting different outfits every day. However, the effectiveness of school uniforms in promoting a positive academic environment is still a point of contention, and their success likely depends on the broader school culture and the way they are implemented.

* Database entries
  + CoPilot
    - Mandatory school uniforms can foster a sense of equality among students by reducing visible socioeconomic differences. When everyone wears the same attire, it minimizes the impact of brand-consciousness or pressure to wear expensive clothing, helping to create a more inclusive environment. This can lead to better interpersonal relationships and reduce bullying based on appearance or financial status, allowing students to focus more on their education.

Additionally, school uniforms promote discipline and a sense of identity within the school community. They encourage students to adhere to a code of conduct and represent their institution with pride. Uniforms can also save time for families in the morning, as children don’t have to decide what to wear each day, making the start of the day less stressful.

However, critics argue that enforcing uniforms stifles personal expression and individuality. Clothing is often a way for students to showcase their personality, and mandatory uniforms can take away this avenue of self-expression. While they may create uniformity, they don’t address deeper issues related to inequality or peer pressure in other aspects of school life.

* + - Opponents of mandatory school uniforms often highlight the importance of individuality and self-expression. Clothing choices allow students to showcase their unique identities, cultural backgrounds, and creativity. Forcing everyone to wear the same outfit may suppress this critical aspect of personal development, especially during the formative school years.

Another argument against uniforms is the potential financial burden they impose on families. Although uniforms are often touted as cost-effective, the specific requirements for certain styles or brands can become expensive. For families with multiple children or limited incomes, this additional expense could be challenging, negating the purported equality uniforms aim to achieve.

On the other hand, advocates of uniforms argue that they can minimize distractions and foster discipline. While these benefits are worth considering, critics believe that schools should instead focus on addressing the root causes of behavioral issues or academic performance, rather than implementing a dress code that may only serve as a superficial solution.

* + - The debate over school uniforms often centers on finding the right balance between equality and individuality. Proponents argue that uniforms create a level playing field by reducing socioeconomic disparities and fostering a sense of community. This can enhance focus on academics and extracurricular activities, as students may be less concerned about their appearance or fashion trends.

Nevertheless, it’s essential to recognize that mandatory uniforms may not suit every school or student population. In creative or diverse communities, allowing students to wear their own clothes could better reflect their individuality and cultural identity. A compromise, such as a relaxed or partial uniform policy, could accommodate the need for both unity and self-expression.

Ultimately, the success of school uniforms depends on thoughtful implementation and feedback from students, parents, and educators. Policies could include options for diverse cultural attire or cost-effective measures for families. By addressing these concerns, schools can aim to achieve a balance that supports both academic excellence and personal growth.

* + GPT 4.0
    - The implementation of school uniforms can promote a sense of discipline and structure within educational environments. When students wear the same attire, it can create a more formal atmosphere that encourages them to take their studies more seriously. Uniforms may also help in reinforcing the idea that school is a place for learning and focus, reducing the likelihood of students getting distracted by casual or inappropriate clothing. This uniformity can foster a sense of belonging among students, making them feel as though they are part of a collective purpose, which may enhance their overall academic experience.

However, some argue that uniforms do not address deeper behavioral issues, such as bullying or student engagement. In fact, by focusing on clothing, schools might overlook other aspects of student well-being, like mental health or personal interests. Moreover, students might still find ways to express themselves outside of clothing, such as through hairstyles, accessories, or behaviors. Thus, while uniforms may reduce certain external distractions, they do not necessarily create a more positive learning environment if the underlying issues remain unaddressed.

In terms of practicalities, uniforms can offer financial benefits for families. They eliminate the need to buy a wide variety of clothes for school, especially those that follow trendy fashions, which can be expensive. Uniforms ensure that all students are dressed appropriately for school without putting undue pressure on families to keep up with the latest fashion. However, this perceived financial relief might be counterbalanced by the costs of purchasing specific uniforms, especially for families with multiple school-aged children or those who need to buy them annually as their child grows.

* + - School uniforms are often seen as a tool for promoting equality and reducing socioeconomic differences among students. In schools where students come from varying economic backgrounds, uniforms help to level the playing field by ensuring that everyone is dressed the same. This can alleviate the pressure to wear expensive or trendy clothes, which is often a source of bullying or exclusion. By eliminating clothing as a status symbol, uniforms can create a more inclusive environment where students are judged based on their character and academic performance rather than their wardrobe.

On the flip side, some argue that uniforms can limit personal freedom and stifle creativity. School is a place where students not only learn academically but also develop their personalities and express their identities. Clothing is a powerful tool for self-expression, and enforcing uniforms may hinder this important aspect of growing up. Critics also point out that uniforms don’t necessarily prevent bullying or cliques; they may just shift the focus to other aspects of difference, such as appearance, mannerisms, or socioeconomic status in other ways that aren't related to clothing.

Despite the drawbacks, uniforms can provide practical benefits for parents, particularly when it comes to simplifying morning routines and making school clothing more affordable. Rather than worrying about outfitting their children with an array of school clothes, parents can buy a set of uniforms that can be worn throughout the year. This can ease the burden of constant shopping and reduce the pressure to spend money on expensive brands or trendy pieces. However, the trade-off is that students may not enjoy the flexibility of dressing in a way that reflects their personal style, which may lead to dissatisfaction.

* + - One of the main arguments in favor of school uniforms is that they can create a sense of unity and reduce the visible distinctions between students. In diverse schools, uniforms help diminish the visibility of wealth or fashion status, potentially reducing bullying or peer pressure. When all students wear the same attire, it can foster a feeling of community and equality, making it easier for students to bond over shared experiences rather than superficial differences. This could lead to a more harmonious school environment where students focus on their studies and relationships rather than competing over appearances.

Despite these potential benefits, there are concerns that uniforms may not be the magic solution to improving student behavior or academic success. Some critics argue that uniforms fail to address underlying issues, such as bullying or poor mental health, that can affect a student's experience in school. They contend that making uniforms mandatory might distract from more significant reforms needed in education, such as improving teaching quality, mental health resources, or addressing safety issues. Additionally, some students may feel restricted by the uniform, leading to frustration and disengagement.

Uniforms could also be seen as a practical way to reduce daily decision fatigue. Students don't have to worry about picking out outfits every morning, which can make the start of the day smoother and more efficient. Parents also save time and money by purchasing a set of uniforms that will last the entire school year. However, this simplicity can come at the cost of students’ sense of autonomy and self-expression. For many, being able to choose what they wear is an important part of developing their independence, and uniforms might undermine that process.

* + Grok AI
    - School uniforms should be mandatory because they promote a sense of equality among students. When everyone wears the same attire, it reduces the pressure to keep up with fashion trends or display socioeconomic status through clothing. This can minimize bullying and social exclusion based on appearance, fostering a more inclusive environment where students focus on learning rather than competing over material possessions.

One potential benefit is the enhancement of school safety. Uniforms make it easier to identify students belonging to a particular school, which can deter unauthorized individuals from blending in on campus. Additionally, they can reduce gang-related conflicts by eliminating clothing that might signal affiliations, creating a more secure atmosphere for education.

However, a drawback is the restriction on self-expression. Students often use clothing as a way to showcase their individuality, and uniforms can stifle this creativity, potentially leading to frustration or disengagement. While the intent is unity, some argue it might come at the cost of personal identity, making it a divisive issue among students and parents alike.

* + - Mandating school uniforms could streamline mornings and reduce decision fatigue for both students and parents. With a set outfit, there’s no need to debate what to wear each day, saving time and reducing stress. This simplicity can help students start their day with a clearer focus on academics rather than wardrobe choices, potentially improving overall productivity.

A key benefit is the fostering of discipline and school pride. Uniforms instill a sense of belonging and collective identity, much like a team uniform in sports. This can encourage students to take their roles as learners more seriously and build a stronger connection to their school community, which might translate into better behavior and academic performance.

On the flip side, the cost of uniforms can be a burden for some families. While proponents argue they’re cheaper than constantly updating a wardrobe, the initial expense—especially for schools requiring specific, branded items—can strain low-income households. This could ironically deepen inequality rather than alleviate it, countering one of the primary arguments for uniforms.

* + - School uniforms shouldn’t be mandatory because they don’t address the root causes of the issues they aim to solve, like bullying or distraction. While they might level the playing field superficially, determined bullies will find other ways to target peers, such as through accessories or verbal harassment. Forcing uniformity might just mask problems rather than resolve them.

A potential benefit, though, is improved focus in the classroom. Without the distraction of flashy or revealing clothing, students and teachers might stay more engaged in lessons. Studies have suggested that uniforms can create a more serious academic tone, which could subtly boost concentration and learning outcomes over time.

Yet, a significant drawback is the lack of evidence proving their effectiveness. Research on uniforms often shows mixed results—some schools report better discipline, others see no change. This inconsistency suggests that mandating them might be an unnecessary imposition, especially if the benefits are anecdotal rather than universal, leaving schools to weigh tradition against practicality.

* + DeepSeek
    - Mandatory school uniforms have been a topic of debate for years, with proponents arguing that they promote equality and reduce distractions. One of the primary benefits is that uniforms can help minimize socioeconomic disparities among students by ensuring everyone dresses similarly, reducing pressure to wear expensive or trendy clothing. Additionally, uniforms can foster a sense of belonging and school pride, creating a more unified and disciplined environment. By removing the focus on fashion, students may also concentrate better on their studies, and uniforms can simplify morning routines for both parents and children.

However, there are also significant drawbacks to mandatory uniforms. Critics argue that uniforms stifle individuality and self-expression, which are important aspects of personal development, especially for teenagers. Forcing students to wear the same clothing every day may lead to resentment or a lack of creativity. Furthermore, uniforms can be costly for families, particularly if they are required to purchase multiple sets or specific brands. Some students may also find uniforms uncomfortable or impractical, which could negatively impact their school experience.

Ultimately, whether school uniforms should be mandatory depends on the specific context and goals of a school community. While uniforms can promote equality and reduce distractions, they may also limit personal expression and impose financial burdens. Schools should carefully weigh these pros and cons and consider alternative approaches, such as dress codes, which balance uniformity with flexibility. Engaging students, parents, and teachers in the decision-making process can help ensure that any policy adopted meets the needs of the entire community.

* I will continue with my testing next class, I will put my text into the AI database next class

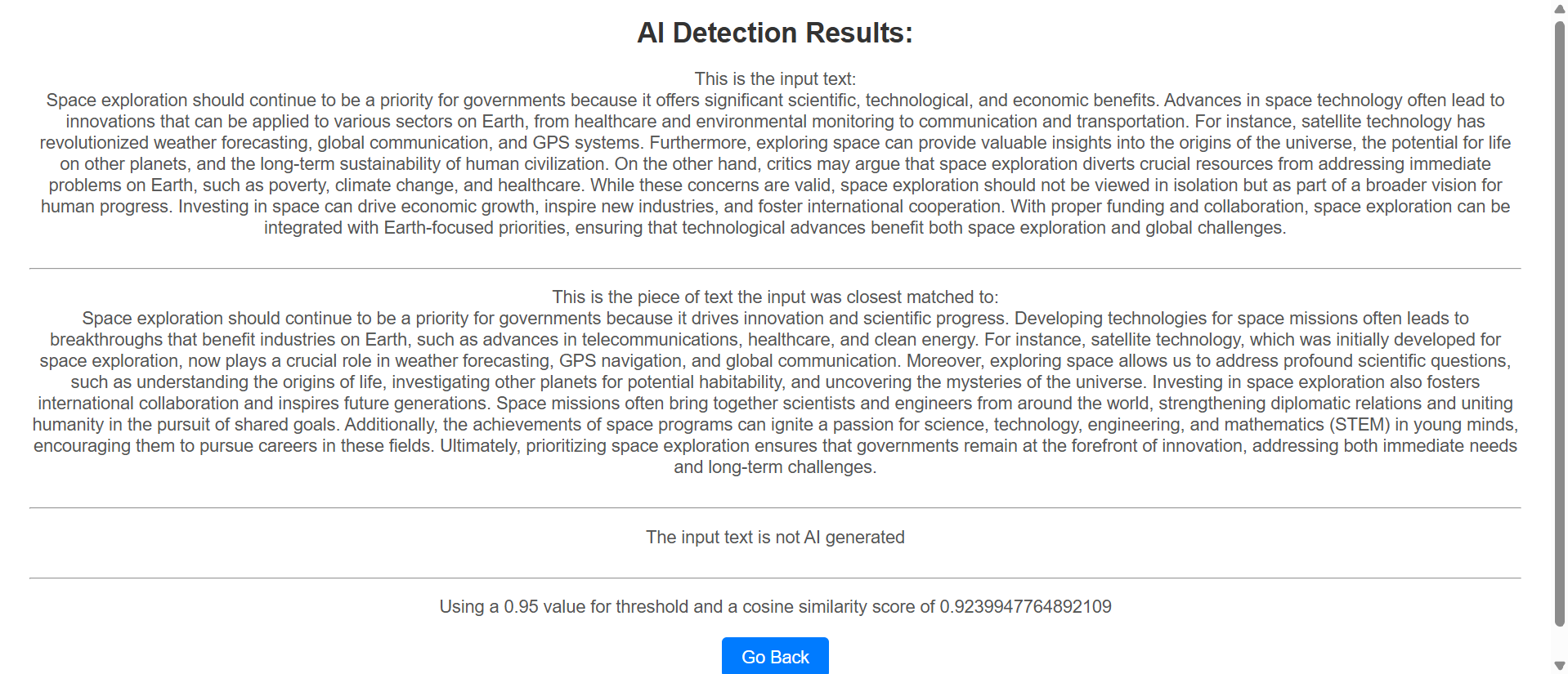
3/20/2025

* First 10-15 mins of class I looked over the final paper and poster deadlines and planned out how I wanted to spend my time for the rest of the year.
* Then I made a GitHub for my project
  + I put my files in, but didn’t completely finish it yet. Will finish it over the weekend
  + Gave access to Dr. Yilmaz and put the link in the slideshow
* Next thing I worked on was changing my GitHub username and name to make it seem more professional
* After that I worked on my project, specifically I have encountered an error with setting up my database, and I have been working to get that error fixed.
  + The database is not generating as it should, so I need to fix that
  + Some entries are not visible inside the database, and that needs to change.
  + <https://www.geeksforgeeks.org/sql-create-table/>
    - This is an article about how to use SQL tables, I am looking this over to try and figure out how to get rid of my error.

3/18/2025

* Testing with a different set of input (started this work on Thursday 3/13)
  + 2 paragraphs instead of 1 paragraph
  + Different Prompt:
    - Should space exploration continue to be a priority for governments? Why or why not?
  + Input Text to test with:
    - “Space exploration should continue to be a priority for governments because it drives technological innovation and addresses long-term global challenges. The advancements made through space programs have already had profound impacts on everyday life, from satellite communications and GPS to medical technologies like advanced imaging systems. These innovations often emerge from the need to solve complex problems in space, which then benefit industries and societies on Earth. Additionally, space exploration helps to expand our understanding of the universe and humanity’s place in it, offering critical insights into the origins of life, Earth's climate systems, and the potential for future human colonization beyond our planet.

Beyond technological advancements, space exploration also plays a vital role in securing humanity’s future. As Earth faces growing challenges such as climate change, resource depletion, and population growth, exploring and utilizing space offers new avenues for resources and energy. For instance, mining asteroids or harnessing solar power from space could provide solutions to Earth’s energy needs. Moreover, the pursuit of space exploration fosters international collaboration, uniting countries with shared goals, and inspiring future generations to push the boundaries of science and engineering. Therefore, investing in space exploration is essential not only for technological progress but for ensuring humanity’s long-term survival and prosperity.”

Results: 

Got a Cosine Similarity around 92% which is very high and is good.

I experimented with putting different sets of text in as input and saw how much they matched the text that was generated by AI.

* Talked with Dr. Yilmaz about my presentation slides and fixed them.

3/17/2025

* Catchup day
  + I worked on revising my presentation slides

3/13/2025

* First 10-15 minutes of class I updated my milestones document to reflect the fact that I was doing testing with manually generated AI results in my database.
* Next I am going to look into my code and try to fix my code for storing in a database. It's giving me a little bit of an error right now.
  + I was able to fix the error
* I talked with Dr. Yilmaz about fixing my code as well.
* After that I started testing my code with a different set of input (I want to see how robust it is when handing different types of text inputs)
* Input Question:
  + Should space exploration continue to be a priority for governments? Why or why not?
* This time I will generate 2 paragraphs of responses for AI as my database
* Responses:
  + GPT
    - **Response 1:** Space exploration should continue to be a priority for governments because it drives technological advancements that benefit life on Earth. Many innovations, such as satellite communication, GPS, and advancements in materials science, have emerged from space research. These breakthroughs have widespread applications that impact industries like healthcare, agriculture, and environmental monitoring. Additionally, space exploration fosters international collaboration and the pursuit of knowledge, allowing nations to come together for the greater good and address universal challenges like climate change.

Moreover, space exploration is essential for the future of humanity. With Earth's resources being finite and the population growing, exploring other planets and celestial bodies may offer opportunities for resource extraction, colonization, or even survival in the event of a global catastrophe. Investing in space exploration today is an investment in the long-term future of humanity, ensuring we are prepared for challenges that may arise beyond our planet.

* + - **Response 2:** While space exploration has provided significant technological advancements, it should not be prioritized over more immediate and pressing issues on Earth. Governments should focus on addressing problems like poverty, climate change, and healthcare rather than spending billions on projects that may not have tangible benefits for people today. The funds spent on space missions could be better allocated to improving quality of life and addressing inequalities worldwide, especially in developing countries where basic needs remain unmet.

Furthermore, space exploration, though exciting, is an expensive endeavor that often benefits large corporations or specific scientific fields rather than the general public. The vast amount of resources required for space exploration could be seen as a luxury when many countries face urgent social, economic, and environmental challenges. While the long-term potential of space exploration is undeniable, governments should first ensure that they are addressing the needs of their citizens before investing heavily in space endeavors.

* + CoPilot
    - **Response 1: For Space Exploration as a Priority** Space exploration should continue to be a priority for governments because it drives innovation and scientific progress. Developing technologies for space missions often leads to breakthroughs that benefit industries on Earth, such as advances in telecommunications, healthcare, and clean energy. For instance, satellite technology, which was initially developed for space exploration, now plays a crucial role in weather forecasting, GPS navigation, and global communication. Moreover, exploring space allows us to address profound scientific questions, such as understanding the origins of life, investigating other planets for potential habitability, and uncovering the mysteries of the universe.

Investing in space exploration also fosters international collaboration and inspires future generations. Space missions often bring together scientists and engineers from around the world, strengthening diplomatic relations and uniting humanity in the pursuit of shared goals. Additionally, the achievements of space programs can ignite a passion for science, technology, engineering, and mathematics (STEM) in young minds, encouraging them to pursue careers in these fields. Ultimately, prioritizing space exploration ensures that governments remain at the forefront of innovation, addressing both immediate needs and long-term challenges.

* + - **Response 2: Against Space Exploration as a Priority** While space exploration is fascinating, it should not take precedence over pressing issues on Earth. Governments face numerous challenges, such as poverty, healthcare crises, climate change, and education inequalities, which demand immediate attention and resources. The high costs of space missions could instead be allocated to addressing these urgent problems. For example, investing in renewable energy initiatives could help combat climate change, while funding public health programs could save countless lives. Prioritizing Earth's challenges over extraterrestrial exploration ensures that governments fulfill their responsibilities to their citizens.

Furthermore, space exploration often involves risks and uncertainties, making it a potentially inefficient use of public funds. Many missions have failed or produced limited returns, raising questions about their overall value. Instead of allocating billions to ventures like Mars colonization, governments could focus on improving life on Earth by advancing sustainable agriculture, water management, and global infrastructure. While space exploration has its merits, it should only be pursued once humanity's most critical needs are met, ensuring a more balanced and responsible approach to resource allocation.

* + DeepSeek
    - **Response 1: Yes, space exploration should remain a priority for governments.**

Space exploration drives technological innovation and scientific discovery, which have far-reaching benefits for humanity. Many of the technologies we rely on today, such as GPS, satellite communications, and advanced medical imaging, were developed or improved through space research. By investing in space exploration, governments foster advancements that not only enhance our understanding of the universe but also improve life on Earth. Additionally, space exploration inspires future generations to pursue careers in science, technology, engineering, and mathematics (STEM), ensuring a skilled workforce for the future.

Furthermore, space exploration addresses existential risks and long-term survival. Earth is vulnerable to threats such as asteroid impacts, climate change, and resource depletion. By exploring space, governments can develop strategies to mitigate these risks, such as asteroid deflection technologies or the potential colonization of other planets. Space exploration also encourages international collaboration, as seen with the International Space Station, fostering peace and cooperation among nations. These efforts underscore the importance of space exploration as a priority for governments, as it safeguards humanity’s future and promotes global unity.

* + - **Response 2: No, space exploration should not be a top priority for governments.**

While space exploration has its merits, governments should focus on addressing pressing issues on Earth, such as poverty, healthcare, education, and climate change. These challenges require significant financial resources and immediate attention, and diverting funds to space exploration could delay progress in solving them. For example, the cost of a single space mission could fund numerous social programs or infrastructure projects that directly improve the quality of life for millions of people. Prioritizing Earth-based issues ensures that governments meet their primary responsibility: serving their citizens and improving societal well-being.

Moreover, space exploration is often driven by geopolitical competition rather than genuine scientific curiosity, leading to wasteful spending and duplication of efforts. Instead of focusing on space, governments could invest in sustainable technologies and environmental conservation to address climate change, which poses a more immediate threat to humanity. While space exploration may offer long-term benefits, it is a luxury that many nations cannot afford, especially developing countries struggling with basic needs. Redirecting resources toward solving Earth’s problems would create a more equitable and sustainable future for all.

Grok

### **Response 1: Yes, Space Exploration Should Remain a Priority**

Space exploration should absolutely continue to be a priority for governments because it drives technological innovation and economic growth. The pursuit of space has historically led to breakthroughs that benefit society as a whole—think of satellite technology, which powers global communication, weather forecasting, and GPS systems. Projects like NASA’s Apollo program or SpaceX’s reusable rockets demonstrate how space ambitions push engineering and science forward, often spilling over into other industries. Governments that invest in space aren’t just chasing stars; they’re seeding advancements that can improve life on Earth, from renewable energy solutions to medical technologies inspired by space research.

Beyond practical benefits, space exploration taps into a fundamental human need to explore and understand our place in the universe. It’s not just about planting flags on Mars or mining asteroids—it’s about answering big questions: Are we alone? Can humanity survive beyond Earth? These endeavors inspire generations, foster international collaboration, and remind us of what’s possible when we think beyond our planet. In a world facing climate change and resource scarcity, space could also offer long-term solutions, like off-planet habitats or resource extraction. Governments should prioritize it because it’s an investment in both our present and our future.

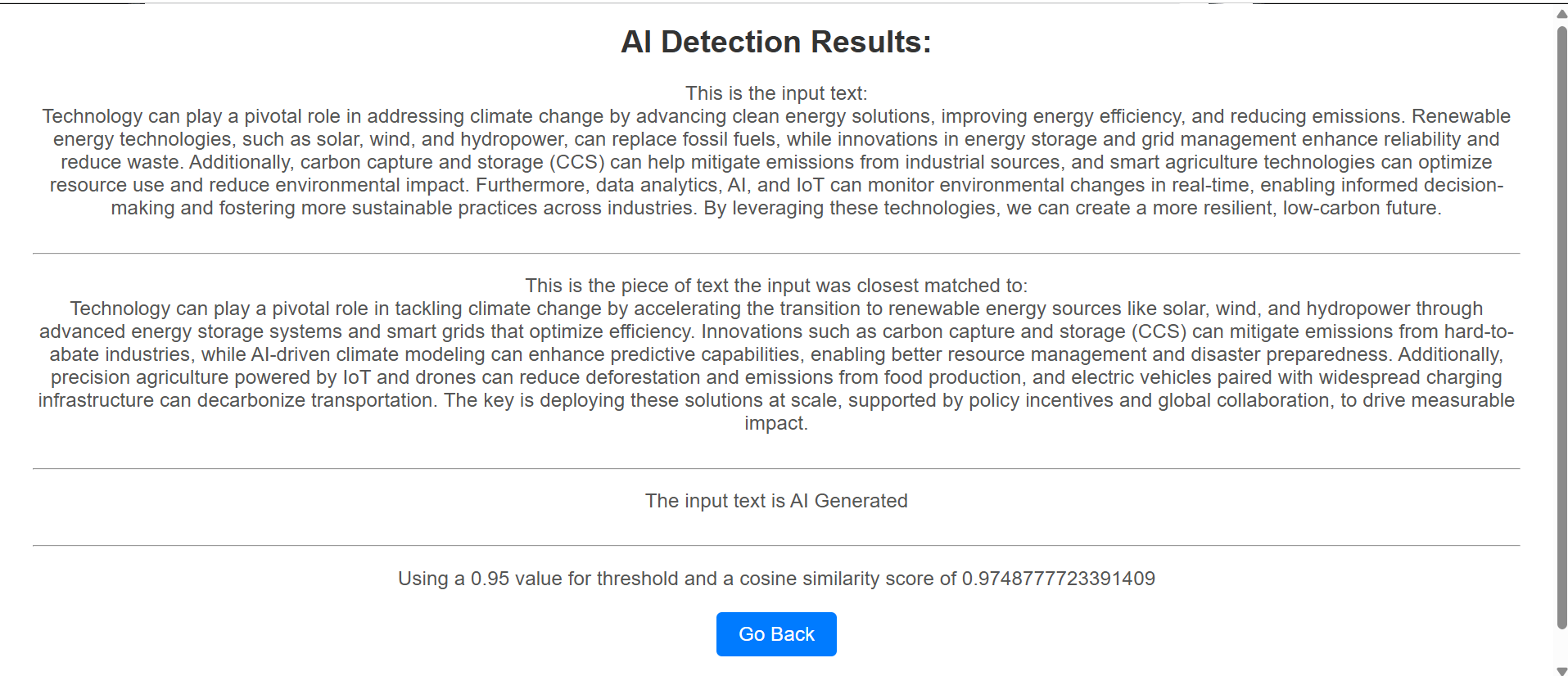
### **Response 2: No, Space Exploration Should Not Be a Priority**

Space exploration should not remain a government priority when there are pressing issues here on Earth that demand attention and funding. Billions are poured into missions to Mars or distant telescopes while millions lack basic necessities like clean water, healthcare, or education. The argument that space tech trickles down to everyday life is shaky—most innovations could be pursued directly without the astronomical (pun intended) costs of launching rockets. Governments should focus on solving climate change, poverty, and infrastructure decay rather than chasing cosmic dreams that benefit only a small elite of scientists and corporations.

Moreover, the risks and uncertainties of space exploration make it a questionable investment. Missions fail, costing taxpayers dearly, and the promise of colonizing other planets remains speculative at best—Earth’s problems won’t vanish by fleeing to Mars. Private companies like SpaceX are already taking up the slack, so why should governments divert resources from immediate human needs to duplicate efforts? Space can wait; it’s been there for billions of years. Right now, the priority should be stabilizing and sustaining the only home we currently have—Earth—before we gamble on the stars.

* Going to use these generations and test them next time.

3/11/2025

* Right now I have gotten results for the cosine similarity part, now what I am going to do is find the highest similarity and use a threshold to compare the most similar entry in the database to the input. If its higher than the threshold, I output as AI generated
  + I am going to code this part right now
* Then after that I will output my results to a different HTML page on the web server
* 
* I got my code to work for a human generated sample of input texts, the ones I picked on Thursday. So I have the basic algorithm part down.
* The way I coded my information, I am using a list and not putting anything into the database right now, so now I am going to code that part.
* Right now I have encountered a bit of an error with storing my list in the database and am working to fix that

3/10/2025

* Continuing to code my manual AI detector
  + Right now I have a problem with comparing the vector of the database and the entry, i'm getting an error and working on solving that error
  + This is the data (I am putting it here for reference so I can use it when I run my code, makes it easier rather than searching for it in last Thursday's entry)
  + Prompt: How should technology be used to address the challenges of climate change?
  + Min word count: 100
  + Max word count: 200
  + Students Text: Technology can play a pivotal role in addressing climate change by advancing clean energy solutions, improving energy efficiency, and reducing emissions. Renewable energy technologies, such as solar, wind, and hydropower, can replace fossil fuels, while innovations in energy storage and grid management enhance reliability and reduce waste. Additionally, carbon capture and storage (CCS) can help mitigate emissions from industrial sources, and smart agriculture technologies can optimize resource use and reduce environmental impact. Furthermore, data analytics, AI, and IoT can monitor environmental changes in real-time, enabling informed decision-making and fostering more sustainable practices across industries. By leveraging these technologies, we can create a more resilient, low-carbon future
  + I was able to solve the issue, now I am getting results for cosine similarity in the 90 percent range which is what I want

3/6/2025

* Last class I limited my search of AI models to GPT 4.5, GPT 4o, GPT 4, GPT 4 turbo
* I can start with GPT 4o and use that for my database
* CHANGE OF PLAN: Now I am going to test my code and detection mechanism against random school questions that I manually got the data from AI generators. I will try and use a bunch of AI generators to test my accuracy.
* Other random questions for later:
  + What makes a good leader, and why?
  + Should school uniforms be mandatory, and what are the potential benefits or drawbacks?
  + How do social media platforms impact teenagers' mental health and relationships?
  + Can art influence social change? Provide examples to support your argument.
  + Should space exploration continue to be a priority for governments? Why or why not?
  + What role does literature play in shaping a society's values and beliefs?
  + Is it better to follow your passion or choose a career for financial stability?
  + How has the internet changed the way we access and value information?
  + What lessons can we learn from history to build a better future?
* Plan:
  + Input Questions: (generated). Start off with 1 paragraph (100-200 words).
    - How should technology be used to address the challenges of climate change?
      * Responses: (AI generated, this is part of the manual database)
        + **Copilot:** Technology should be harnessed strategically to combat climate change through innovation and efficiency. Renewable energy technologies, such as solar panels and wind turbines, are vital for reducing dependence on fossil fuels and cutting greenhouse gas emissions. Smart systems, like energy-efficient grids and AI-driven tools, can optimize resource use and minimize waste in industries and homes. Advances in carbon capture and storage can help reduce existing atmospheric CO2 levels, while precision agriculture and water-saving irrigation technologies can support sustainable food production. Furthermore, data analytics and satellite monitoring can track climate patterns and provide actionable insights for effective policymaking. By embracing these technologies, we can pave the way for a sustainable and resilient future.
        + **Copilot:** Technology offers powerful tools to tackle the multifaceted challenges of climate change. Renewable energy innovations, such as offshore wind farms and advanced battery storage systems, can ensure a steady supply of clean energy while minimizing environmental harm. Green transportation technologies, including electric vehicles and hydrogen-powered systems, reduce emissions and transition us away from fossil fuels. Breakthroughs in materials science, like energy-efficient building materials and biodegradable products, can lower carbon footprints in construction and manufacturing. Furthermore, artificial intelligence and machine learning can predict climate trends, identify risk zones, and optimize resource allocation. By integrating these advancements into everyday practices and policies, we can mitigate climate impacts and build a sustainable future.
        + **GPT : \*\*Response 1:\*\***
        + Technology should be leveraged to address climate change by advancing renewable energy sources like solar, wind, and hydropower, reducing dependency on fossil fuels. Innovations such as energy storage solutions, carbon capture technologies, and electric vehicles can accelerate the transition to a low-carbon economy. Additionally, AI and data analytics can improve energy efficiency, optimize supply chains, and provide real-time climate modeling to guide policy decisions. By fostering collaboration between governments, industries, and innovators, technology can be a powerful tool to mitigate climate impacts and promote sustainable development.
        + **\*\*Response 2:\*\***
        + To combat climate change effectively, technology must focus on both reducing emissions and adapting to changing environmental conditions. Smart grids and energy-efficient systems can lower energy consumption, while next-gen agricultural technologies can enhance food production with minimal environmental impact. Furthermore, digital tools can monitor deforestation, pollution, and wildlife health, enabling faster, more informed responses. Investments in research, innovation, and infrastructure development are essential to ensuring that technology supports a global shift toward sustainability and climate resilience.

**DeepSeek**

* + - * + \*\*Response 1:\*\*
        + Technology should be leveraged to address climate change by accelerating the transition to renewable energy sources, such as solar, wind, and geothermal power, through innovations in energy storage, grid management, and efficiency. Smart technologies, like IoT-enabled devices and AI-driven systems, can optimize resource use, reduce waste, and monitor environmental changes in real time. Additionally, carbon capture and storage (CCS) technologies can help mitigate emissions from industries that are harder to decarbonize. By investing in sustainable agriculture, electric vehicles, and circular economy models, technology can drive systemic change, enabling societies to reduce their carbon footprint while fostering economic growth and resilience.
        + \*\*Response 2:\*\*
        + Technology plays a critical role in combating climate change by enabling data-driven solutions and scalable innovations. Advanced climate modeling and satellite monitoring can improve our understanding of environmental changes, helping policymakers make informed decisions. Renewable energy technologies, such as next-generation solar panels and offshore wind farms, can replace fossil fuels, while energy-efficient buildings and smart cities can reduce overall consumption. Biotechnology can also contribute by developing climate-resilient crops and carbon-sequestering materials. However, equitable access to these technologies is essential to ensure that developing nations can also participate in and benefit from the global transition to a low-carbon future.
        + **Grok**
        + Response 1: Technology can play a pivotal role in tackling climate change by accelerating the transition to renewable energy sources like solar, wind, and hydropower through advanced energy storage systems and smart grids that optimize efficiency. Innovations such as carbon capture and storage (CCS) can mitigate emissions from hard-to-abate industries, while AI-driven climate modeling can enhance predictive capabilities, enabling better resource management and disaster preparedness. Additionally, precision agriculture powered by IoT and drones can reduce deforestation and emissions from food production, and electric vehicles paired with widespread charging infrastructure can decarbonize transportation. The key is deploying these solutions at scale, supported by policy incentives and global collaboration, to drive measurable impact.
        + Response 2: To address climate change, technology should focus on enhancing sustainability across sectors, starting with scalable clean energy solutions like next-generation nuclear reactors and geothermal systems that provide reliable, low-carbon power. Biotechnology can contribute by engineering carbon-absorbing plants or microbes, while advancements in materials science can produce eco-friendly alternatives to plastics and concrete, major emissions culprits. Digital tools, such as blockchain for transparent carbon credit trading and satellite monitoring for real-time emissions tracking, can enforce accountability and optimize resource use. Success hinges on integrating these technologies with equitable access and funding, ensuring they benefit both developed and vulnerable regions alike.
  + Next step is I put these responses into the database I have and encode them
  + Sample Response to test against:
    - Technology can play a pivotal role in addressing climate change by advancing clean energy solutions, improving energy efficiency, and reducing emissions. Renewable energy technologies, such as solar, wind, and hydropower, can replace fossil fuels, while innovations in energy storage and grid management enhance reliability and reduce waste. Additionally, carbon capture and storage (CCS) can help mitigate emissions from industrial sources, and smart agriculture technologies can optimize resource use and reduce environmental impact. Furthermore, data analytics, AI, and IoT can monitor environmental changes in real-time, enabling informed decision-making and fostering more sustainable practices across industries. By leveraging these technologies, we can create a more resilient, low-carbon future.
  + Code it to work and see the results
    - Cosine similarity
    - Threshold
    - Output AI generated or not

3/4/2025

* Continuing research into different openAI models that are relevant for using
* First thing I am doing today is looking into different AI models in open AI for the best one for my database.
  + <https://www.datacamp.com/tutorial/openai-o1-api>
    - This is a newer GPT model (o1) it gives more advanced answers, but takes longer time to generate a response.
    - This doesn’t seem widely used, and is really expensive so I am not going to use this one
    - O1 and o3 are reasoning models, that means that they think before they generate a response. These aren’t really commonly used by students because students want instant results, so I am not using this model or o3.
  + <https://www.datacamp.com/tutorial/gpt4o-api-openai-tutorial>
    - This is an article about GPT 4o
      * Used more commonly by everyday people
      * Might be better for my purpose
* Research Questions:
* First question I need to answer is what AI model am I using to generate my database? (this is why I am researching into openAI models)
  + - https://platform.openai.com/docs/models#:~:text=o3%2Dmini%20is%20our%20most,%2C%20Batch%20API%2C%20and%20more
      * GPT 4.5
      * GPT 4o
      * GPT 4
      * GPT 4 turbo
* Second question I need to answer is how does the pricing work for that model, what do I need to buy and how will I feed my question into the AI model in order to minimize the cost.
  + - <https://openai.com/api/pricing/>
      * This is the openAI page that relate to pricing
* Third question I need to answer is: how do I write the code to access that model and get the responses back to my code.
  + - I need to understand the details of how to use the API for the AI generator I choose to use
    - Will continue research on this next class

3/3/2025

* I’ve realized that I have to have a more modern approach, so instead of GPT 4.0 I will try using GPT 4o. That is the newer version of GPT 4.0 and is probably more well used by people that use generative AI.
* I might need a subscription to get GPT 4o, but I can pay for that if needed. I think right now in order to use GPT 4o we need to have a subscription.
* <https://www.datacamp.com/tutorial/gpt4o-api-openai-tutorial>
  + This is a tutorial to use gpt 4o API
  + I will try using this now instead of GPT 4.0
  + Should align more with modern uses of AI
  + I am reading through this article and seeing if it can help me, specifically if I can avoid the openAI is out of date error
* <https://www.datacamp.com/tutorial/openai-o1-api>
  + Also have this article about o1, which a different OpenAI
  + Talks about how to install and use o1.
  + I am reading this article and will use it to help me to send questions and get responses from GPT
* <https://platform.openai.com/docs/models#:~:text=o3%2Dmini%20is%20our%20most,%2C%20Batch%20API%2C%20and%20more>.
  + I am also looking at this. This article talks about all of openAI’s models. I will look over all the models openAI has and need to choose one in order to generate my database, I might use multiple of them. I am reading this to know what my options are in terms of which AI I can use to make my database.
* <https://platform.openai.com/docs/guides/reasoning>
  + This is more from openAI’s website about reasoning models and when they are used

2/27/2025

* Still trying to figure out how to appropriately hook up my code to GPT 4.0 (or newer AI model)
  + <https://help.openai.com/en/articles/7102672-how-can-i-access-gpt-4-gpt-4-turbo-gpt-4o-and-gpt-4o-mini>
    - I am now looking at the OpenAI website, as the other websites didn’t really have sufficient information to get access to GPT 4.0.
    - I might have to use a paid version in order to access GPT 4.0 or GPT4o (the newer version)
  + <https://levelup.gitconnected.com/build-your-own-question-answering-system-with-openai-and-flask-2200507ac601>
    - Now I am looking into this article, it talks about using a flask server and using openAI with it, I will try coding this and see if it doesn’t have the same error.
    - All the websites I am experimenting with to try and use part of their code, they all have the problem of being outdated. I need to figure out a solution to that problem somehow.
  + <https://cloud.google.com/endpoints/docs/openapi/when-why-api-key>
    - I also looked into this article, to understand API keys a little bit better
    - Helps for my OpenAI key that I have
* I also refined my neural network a little bit, I improved it and changed some more parameters around
* Now I am thinking of some alternative options to feeding OpenAI because it's not working right now. I will try a little bit more to make this error go away, but I'm just not 100 percent sure how to fix the error.
  + I might have to take a more step by step approach and do one step at a time and see that work

2/25/2025

* First part of class I did my own presentation
* Second part of class I continued to work on solving my problem of OpenAI being outdated
  + <https://github.com/openai/openai-python/discussions/742>
    - Using this website
  + Im testing my openAI part, still isn’t 100 percent working yet

2/24/2025

* Starting off where I left off last class, trying to find a way to solve the problem of the OpenAI being outdated
* <https://github.com/openai/openai-python>
  + This is the website that VisualStudio gave me to try and fix it, so I am looking on it right now to find the answer to the error
* <https://github.com/openai/openai-python/discussions/742>
  + This link also seems helpful in making my code work. I am reading through it to find a solution and implement that in my code.

2/20/2025

* Over the last weekend I wrote some code to take my question and sent it to Chat GPT 4.0, where I could get responses and bring those back to my server. I tried an approach on <https://genezio.com/blog/build-chatgpt-chatbot-flask/#step-1-install-flask-and-required-libraries> but that is still giving me an error right now, I am currently trying to debug that error and make it work
* <https://careerfoundry.com/blog/web-development/what-is-npm/#:~:text=npm%20stands%20for%20Node%20Package,over%2011%20million%20developers%20worldwide>.
  + I tried looking into this website to debug my error and figure out why npm install -g npm this doesn’t work
  + Couldn’t really find the answer here
* I am trying to change my approach to go one step at a time
  + First step: try and take the question the user entered and put it into ChatGPT 4.0
  + <https://www.geeksforgeeks.org/implement-chatgpt-in-a-flask-application/>
    - I will start with trying to implement the code on this website, but for my purpose
    - I have coded this, but the problem is that I am not getting a response from Chat GPT as I expected, so now I have to look at the code/website and figure out what is wrong.
    - I think the version of OpenAI they were using is outdated, so I have to change the code to make it the new version.
    - Currently looking into ways to fix this error of OpenAI being outdated
    - I was looking around and tried to edit my code to fix this error, but I haven’t been able to fully finish it yet. I will continue to look into this error of OpenAI being outdated and find a solution for it.

2/18/2025

* Senior Skip Day - I showed up and Free Day

2/13/2025

* First I tested my website a little bit more with different inputs to make sure it works
* <https://genezio.com/blog/build-chatgpt-chatbot-flask/#step-1-install-flask-and-required-libraries>
  + I am looking at this article and how to integrate ChatGPT 4.0 to my project. First I will look at the code they have provided and run that on my own application to see that it works (reading through what they said and testing that), then later I will take what they wrote and tail it to my needs so that it does what I want.
  + I'm looking over the website and trying to learn why the code is written how it is so that I can change the code around when I use it, I will need to modify it to an extent.
  + Right now I am encountering a problem with installing a library, so I am debugging that and trying to solve that issue.
* After this, I started to look at <https://www.geeksforgeeks.org/how-to-choose-batch-size-and-number-of-epochs-when-fitting-a-model/>. So that I can get some information on how many epochs and batch size I can use on my Doc2Vec model to achieve good performance. I need to experiment with this more in order to get the best results.

2/11/2025

* <https://www.sciencedirect.com/topics/computer-science/cosine-similarity#:~:text=Cosine%20similarity%20measures%20the%20similarity,document%20similarity%20in%20text%20analysis>.
  + Today I read this article to familiarize myself with how cosine similarity works
  + This document gives a simple example of how cosine similarity can be used with documents
  + I wanted some more background to strengthen my understanding of how Cosine Similarity works and how I can implement it in my website
* Then I started coding up my website to send the input question to chatGPT 4.0, this seems like a long process but I just started writing the initial code based on the websites that I have read on

2/10/2025

* <https://shwinda.medium.com/build-a-full-stack-llm-application-with-openai-flask-react-and-pinecone-part-1-f3844429a5ef>
  + Now I’ve started to look into this article, about using LLM with a Flask application (which I am building as part of my project)
  + I started to read this article during the class. I didn't have that much time in class so I just read, and didn't do a lot of other things in terms of code.
  + My end goal is to take the question the teacher puts in, put that question into GPT 4.0 and then take the output and put that back into my code and encode that as part of my database.
    - This will be used to predict whether its GPT 4.0 generated or not

2/6/2025

* <https://www.youtube.com/watch?v=GekTersI_Cs>
  + I started to look into this video about integrating ChatGPT 4.0 into specifically a python flask server
  + Started to watch the video
  + Seems like he walks through it step by step so that I can understand it, will try his approach and see where it gets me
* <https://medium.com/@shikhararyan/integrating-gpt-4o-api-with-your-chat-interface-to-handle-text-and-image-inputs-130df58e8850>
  + I also opened up this article to look into, I need to read more into these resources to figure out how to incorporate GPT 4.0 into my website so that I can feed it prompt questions and it will output responses that I can store
* <https://genezio.com/blog/build-chatgpt-chatbot-flask/>
  + Also looked into this article
* Will look into these articles a little bit more later and try to use them to hook up GPT to my website

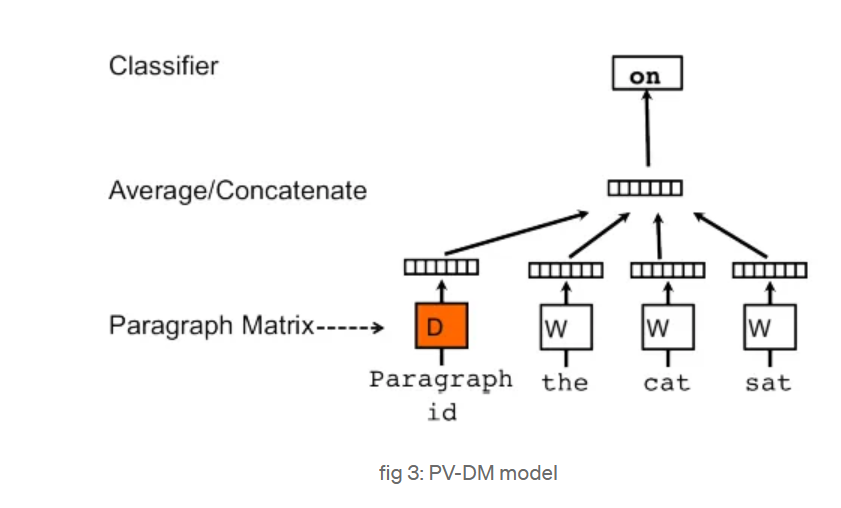
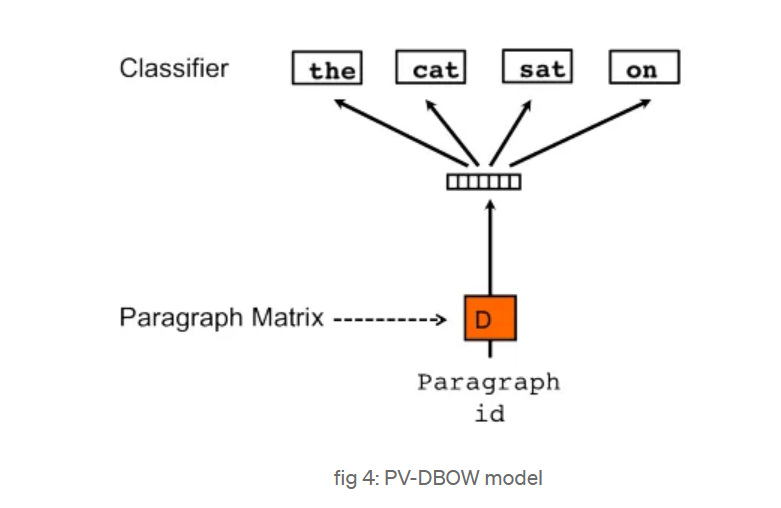
2/4/2025

* First couple of minutes I read over my presentation again.
* <https://blog.hubspot.com/website/chatgpt-integration>
  + Now I am looking into this website to find how to integrate ChatGPT into my website.
  + I want to feed the prompt to ChatGPT so that it can give me responses back and I can use those encoded responses in my database
  + I took a temporary break from the ChatGPT part and added another question to my website. This question asks the user the minimum word amount and maximum word amount in the essay/piece of writing.
  + I looked over this website, and while reading it, I am trying to go step by step and integrate ChatGPT into my website.
    - Step 1: make an API Key
    - Step 2: (this article started talking about integrating GPT into a web server that I do not know how to code, so I decided to use another article and try that)
  + <https://mosquid.medium.com/using-chatgpt-as-a-backend-8e675094e7b4>
    - Started reading this site and looked into how they want to code the backend of ChatGPT into their website, I might have to do some research to find the right article to guide me in using ChatGPT as a backend for my project

2/3/2025

* I read through my slides and made sure that everything looked good and was updated to present, I also read through my milestones to make sure that they looked good as well.
* I then started to work on my HTML website, and putting a prompt into the website where the user can specify the question that the essay was written about, this will then go to ChatGPT 4.0 and generate responses for my database.
* Tried to edit the CSS of my website so that it looked more visually appealing, especially the second page.

1/30/2025

* First part of class I used LucidChart to improve my chart design on my presentation
  + I did this using Canva, now the look of the chart is much nicer, the arrows are straight and it looks more clean and better presented than it did before. I have it saved for if I want to change anything up later.
* Second part of class I considered using Doc2Vec instead of Word2Vec as that might work better for analyzing essays and longer pieces of text than Word2Vec, Word2Vec might make this too complicated
  + <https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e>
  + I am reading this article to start and get a background on Doc2Vec and how it works in general
  + Numerical representation of text document
  + Length of the document doesn't matter, it can be used for longer texts as well
  + Very similar to Word2Vec, but there is an additional Paragraph Vector that is added on (unique to each document)
  + This document vector is trained and at the end you come up with a representation of the document as a whole
  + 
  + This model architecture is called Distributed Memory version of Paragraph Vector (PV-DM)
  + Keeping the context of the document in mind
  + Another algorithm which models skip-gram is also used
  + Distributed Bag of Words version of Paragraph Vector (PV-DBOW)
  + 
  + Its very fast because it doesn’t save the word vectors
  + Authors think that using combination of both algorithms would be effective
  + PV-DM is better, and can give very good results by itself
  + To train: need a set of documents
  + Vector D is made for each document

1/28/2025

* First part of class I expanded my slides on encoding and am going to show that to Dr. Yilmaz in order to earn back credit on it
  + <https://swimm.io/learn/large-language-models/what-is-word2vec-and-how-does-it-work>
  + I looked on this website in order to improve my slides and get an image
* Second part of class I researched more into Word2Vec and how I can use that in my code for encoding the database entries and the input entries
* Towards the end of class I talked with Dr. Yilmaz more on how to improve my presentation and got some suggestions for future feedback.

1/23/2025

* I revisited the feedback I got on my project presentation and edited my slides, also looked at the journal and milestones and edited those
* I also worked more on my word2Vec encoding and looked at different websites that had different approaches to how to use word2Vec best for my application of semantic similarity and AI detection

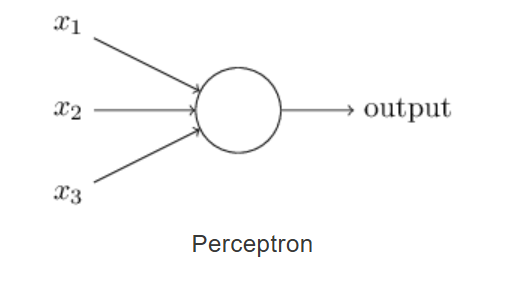
1/21/2025

* Took the code from <https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/> and downloaded the files and ran the code on my own laptop
* Now it is giving me an error saying that it cannot find the directory which the file is inside
  + I am trying to debug this error right now
  + Will continue working on it next class
* I copied the file path but it’s still giving me an error
* Also started to read this: <https://www.kaggle.com/code/pierremegret/gensim-word2vec-tutorial> which helps me understand how Gensim Word2Vec model actually works
  + It's a tutorial on how to use Word2Vec with Gensim

1/16/2025

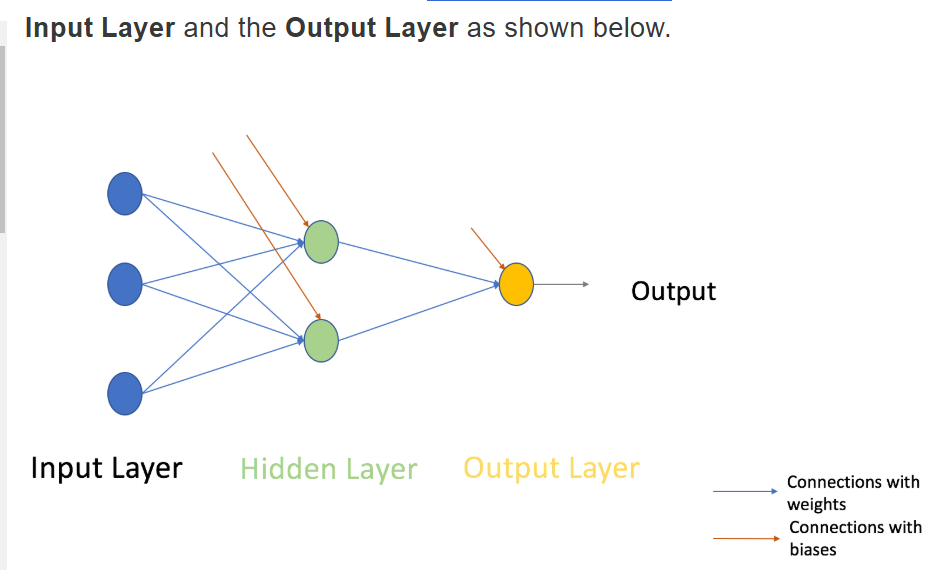
* <https://www.analyticsvidhya.com/blog/2020/07/neural-networks-from-scratch-in-python-and-r/>
  + Full Batch Gradient Descent and Stochastic Gradient Descent
    - Change/update the weights in the multilayer perceptron
    - Difference between the two methods of gradient descent is the way they actually update weights and biases
    - Full Batch Training Algorithm: takes all of the training data points and updates all of the weights one time
    - Stochastic Gradient: Takes 1 or more than 1 sample but doesn’t take all of the training data in order to update the weights one time
  + Steps involved in neural network methodology
    - Steps in using NN
      * Take in the input and output
        + X > input
        + Y > output
      * Make the weights and biases random values (one time)
      * Matrix dot product
  + I will finish looking at this article later, I think I have enough background information right now
* <https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/>
  + Now I am reading and looking into this article for the Word2Vec encoding
  + Starting to encode and test what this website said, I'm not sure if it will work but going to try it at least

1/14/2025

* First thing I did was rerun and test my code so that I knew exactly what it did.
* Now i'm looking back over all the websites on Word2Vec encoding in order to remember how I want to encode my project
* <https://carpentries-incubator.github.io/python-text-analysis/09-wordEmbed_train-word2vec/index.html>
  + I am looking at this website specifically and how to take in the paragraph from the website and separate the paragraph into sentences, then take the sentences and go word for word through it.
  + <https://www.analyticsvidhya.com/blog/2020/07/neural-networks-from-scratch-in-python-and-r/>
    - I have now started looking at this website to get a better intuition of how neural networks work and how I can apply the neural network to my problem of Word2Vec encoding
    - Several inputs go through multiple neurons from multiple hidden layers and return the result from the output layer. This is forward propagation
    - Compare the result given from the code to the expected result
    - Back propagation: Going back through the neural network and trying to reduce the error as much as possible > using gradient descent
    - Perceptron: Take in multiple inputs and gets out one output
    - 
    - The actual input-output relationships:
      * Directly combine the input and set a threshold. So if the sum of the input is greater than the threshold, the resultant output is one and if it's not, then the resultant output is zero.
      * Adding weights to the inputs
        + Multiply the weight corresponding to each input by the input and add them all together. Ex: W1 \* I1 + W2 \* I2 + W3 \* I3. Check if this value is greater than the threshold set.
        + In this case the weight can determine how important a single input node is. The bigger the weight, the more importance it has.
      * Adding in bias
        + How flexible the perceptron is
        + Another weight that is assigned to the bias, its like the +b in a linear equation
        + Linear representation: w1\*x1 + w2\*x2 + w3\*x3 + 1\*b. Where b is the bias
        + This is linear, nowadays the way to calculate a perceptron is through non-linear ways
      * Activation function
        + Takes in the sum of the weighted outputs, and returns the output of the neuron
        + “Sigmoid”, “Tanh”, “ReLu” are examples of some activation functions
      * Fwd prop, Backward prop, epochs
        + When you have high error (output is far from the expected output), it is necessary to backwards propagate. Change the biases and weights based on the error.
        + Find the error and propagate it back into the neural network
        + To minimize the error

Find the gradient (derivatives) of each node with respect to the final output

One time forward propping and going backwards propping after that is called an epoch

* + - * Multilayer perceptron
        + Hidden layers between the input layer and the output layer
        + 
        + Fully Connected: Every node in a layer is connected to every node in the previous layer and the next layer with the exception of the input and output layer.
        + Gradient descent is most commonly used for training

1/13/2025

* Reread over my code and familiarized myself with the code again as it's been a long time since I have looked at it.
* Edited my code and tried to continue my word vector encoding

12/19/2024

* Continued work on my word vector encoding
  + <https://medium.com/@manansuri/a-dummys-guide-to-word2vec-456444f3c673#:~:text=Word2Vec%20embeddings%20can%20be%20trained,free%20open%20source%20python%20library>
  + <https://colab.research.google.com/drive/1YkSrvfWR_EBFFrhV5E15Z6k5es4Kluom?usp=sharing>
  + <https://carpentries-incubator.github.io/python-text-analysis/09-wordEmbed_train-word2vec/index.html>
  + I am trying to code something similar to these websites code, but it hasn’t been worked so far
  + <https://www.tensorflow.org/text/tutorials/word2vec>
    - I am looking into this website now, trying to figure out more about how to code up the Word2Vec so that it works for my purpose of identifying ChatGPT written code
    - I think I might need some more background information on Word2Vec so that I can explain it better
    - Word Embeddings > Large Datasets
    - Good for NLP
    - Continuous Bag of Words Model
      * Figures out what the word in the middle would be based on the words around that middle word
      * Order of the words is not relevant in this specific model
    - Continuous Skip-gram
      * Does the opposite of the continuous bag of words, predicts the words surrounding a current word, using the content of the current word.
    - Skip-gram and negative sampling
      * Taking in a word and trying to figure out the words around that word
      * Use Skip grams
        + “which are n-grams that allow tokens to be skipped”

12/17/2024

* Worked on my word vector encoding
  + I am currently reading at and trying to implement parts of what is written on these websites
    - <https://medium.com/@manansuri/a-dummys-guide-to-word2vec-456444f3c673#:~:text=Word2Vec%20embeddings%20can%20be%20trained,free%20open%20source%20python%20library>
    - <https://colab.research.google.com/drive/1YkSrvfWR_EBFFrhV5E15Z6k5es4Kluom?usp=sharing>
    - https://carpentries-incubator.github.io/python-text-analysis/09-wordEmbed\_train-word2vec/index.html

12/16/2024

* I was in class - Senior Skip Day

12/12/2024

* Did my own presentation in the middle of class
* Continued to work on my Word2Vec encoding
  + Looking into one hot encoding and how to train my model
  + Continued to troubleshoot the errors in my code

12/10/2024

* First part of class I looked over my presentation
* Second Part of class I reviewed this: <https://medium.com/@manansuri/a-dummys-guide-to-word2vec-456444f3c673#:~:text=Word2Vec%20embeddings%20can%20be%20trained,free%20open%20source%20python%20library>. And tried to find a way to encode the paragraphs that will be put into the database. I started writing code for encoding the entries into the database, but it doesn’t fully work right now and I am encountering some errors that I need to fix.

12/9/2024

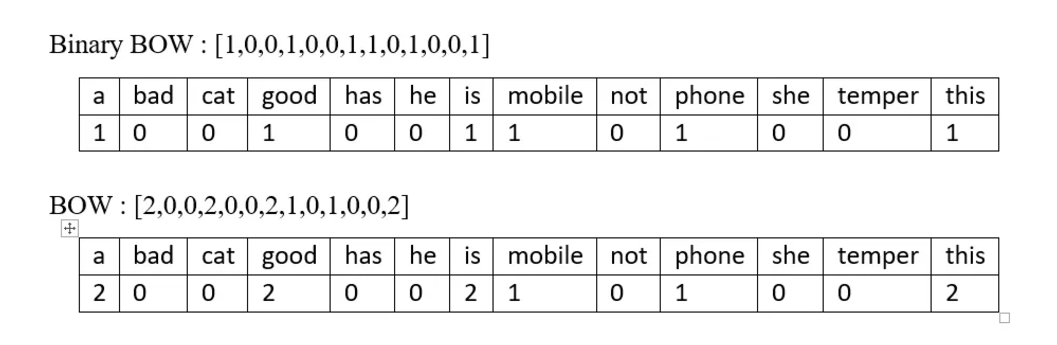
* Listened to presentations

12/5/2024

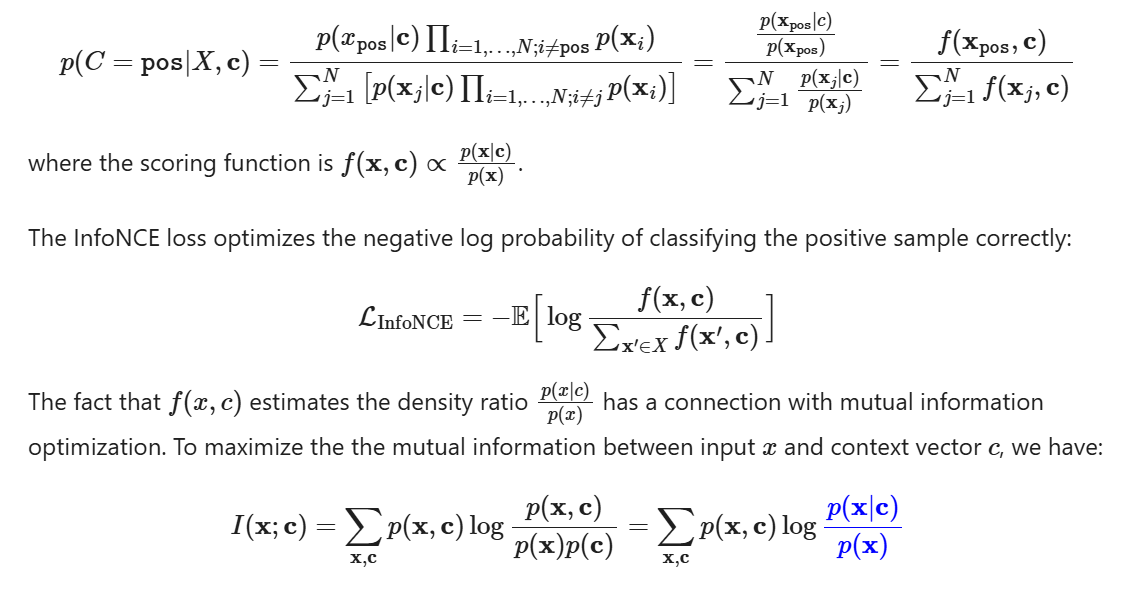
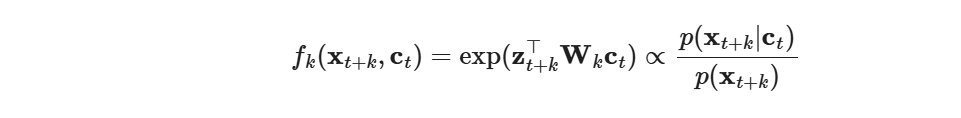
* We listened to presentations

12/3/2024

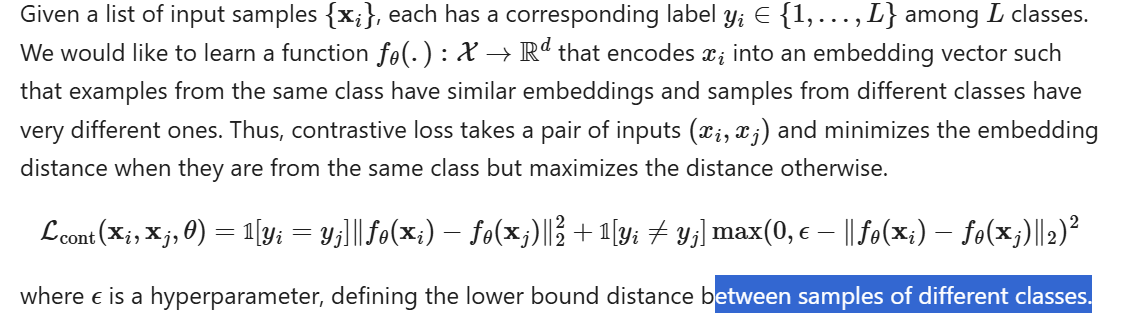
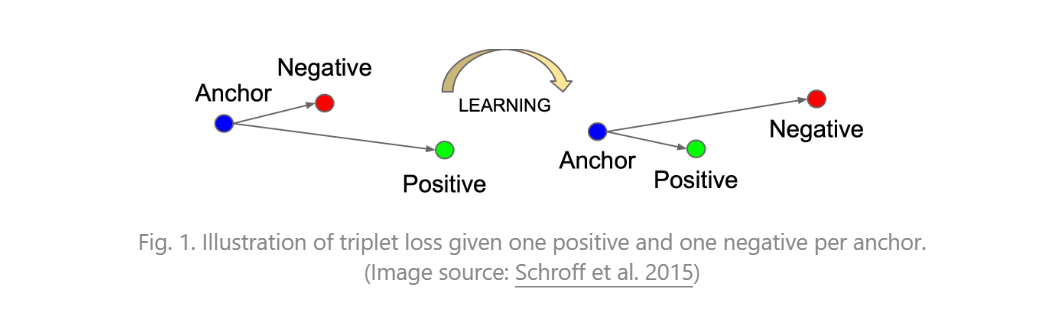
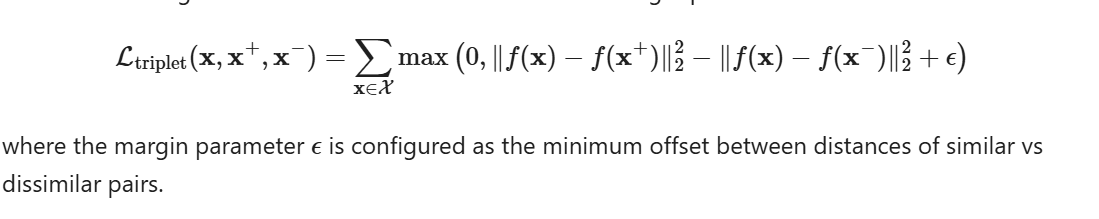
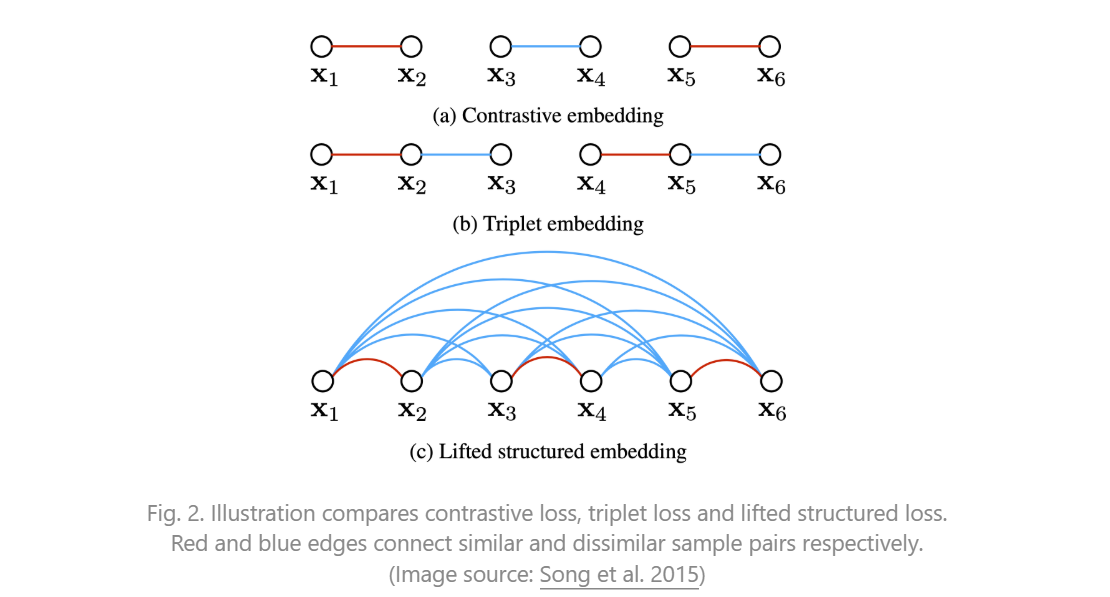
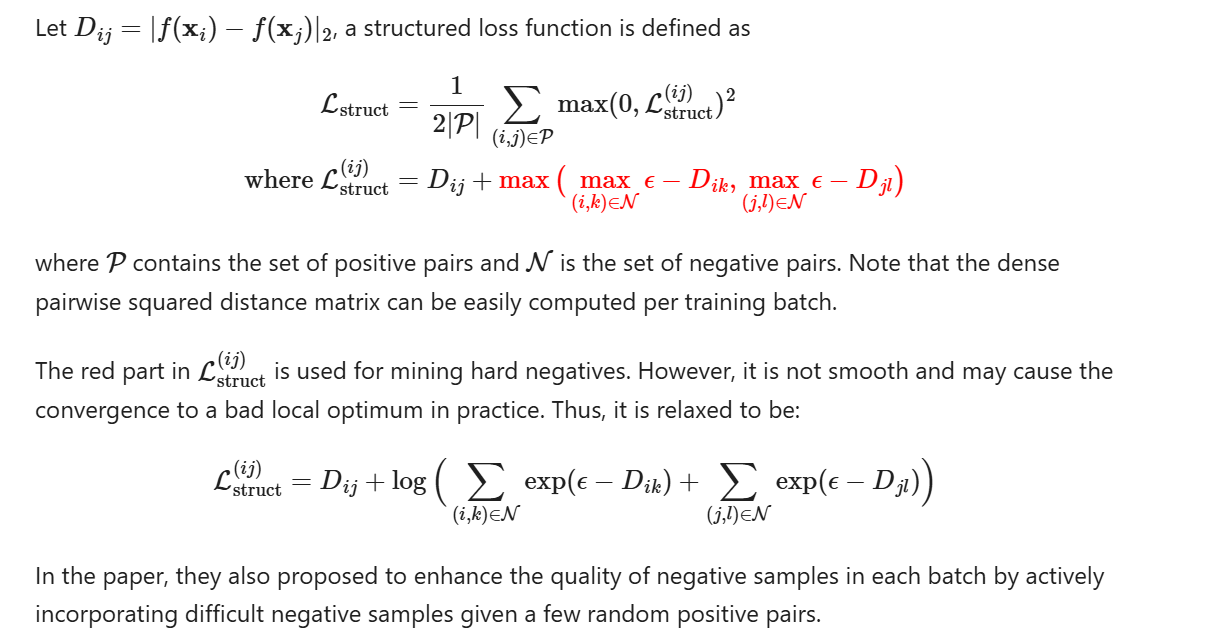
* Spent most of the time working on the presentation
* NLP - Text Encoding: A beginners guide (looking more into the specifics of how to encode the text)
  + <https://bishalbose294.medium.com/nlp-text-encoding-a-beginners-guide-fa332d715854>
  + The whole premise is that machines don’t know what sentences or characters are, they only understand numbers
    - This is the reason for text encoding
  + Strategies to encode
    - Index-Based Encoding
    - Bag of Words (BOW)
    - TF-IDF Encoding
    - Word2Vec Encoding
    - BERT Encoding
  + Document Corpus: All the text that will be used
  + Data Corpus: All the words that are unique in the document corpus
  + **1. Index Based Encoding**
    - Index each of the words in the data Corpus from 1 to N, each word is given a unique index
    - In the Document Corpus, replace the text with the index of each of the words
    - At this point, each word has an index with it. Each sentence is a list with indexes in it. Those lists together make up the Document Corpus.
    - Need to use Max Padding
      * Find the longest sentence from the Document Corpus, and make all the other sentences that long. Ex: if the longest sentence is 6 words long, make all the other sentences 6 words long as well. Use the index zero for padding, add 0’s to the end of all the sentences until they have the same length as the longest sentence
  + **2. Bag of Words (BOW)**
    - Making use of all of the data corpus to encode the sentences
    - Represent a sentence based on the words it does and doesn’t have from the data corpus.
      * If it is in the data corpus, put a one in that place
      * If it isn’t in the data corpus, put a zero in that place
    - Can do this for all the sentences, each sentence is represented by the data corpus (all the words, without repeats) and zeros and ones are filled based on whether the word is in the sentence or not
    - Binary BOW
      * Only uses 1 or 0 for each word in the sentence
      * Does not take into account the count of each word in the sentence
    - BOW
      * Also look at how many times a word was in a sentence (frequency)
      * Provides information on the relative amount of a word in a sentence, not just whether a sentence has a word or not
    - BOW does not look at the sequence of the sentences

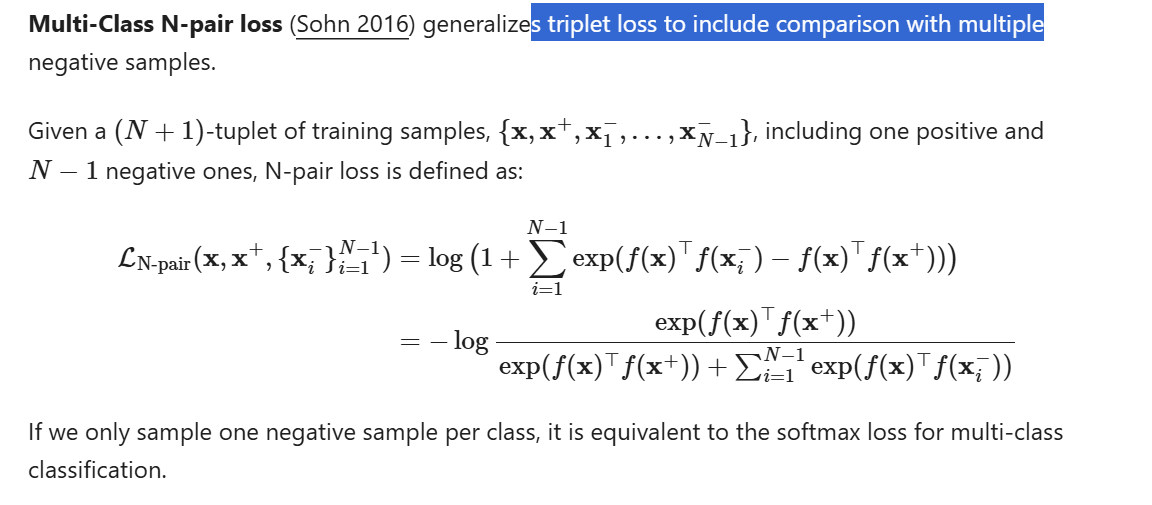


12/2/2024

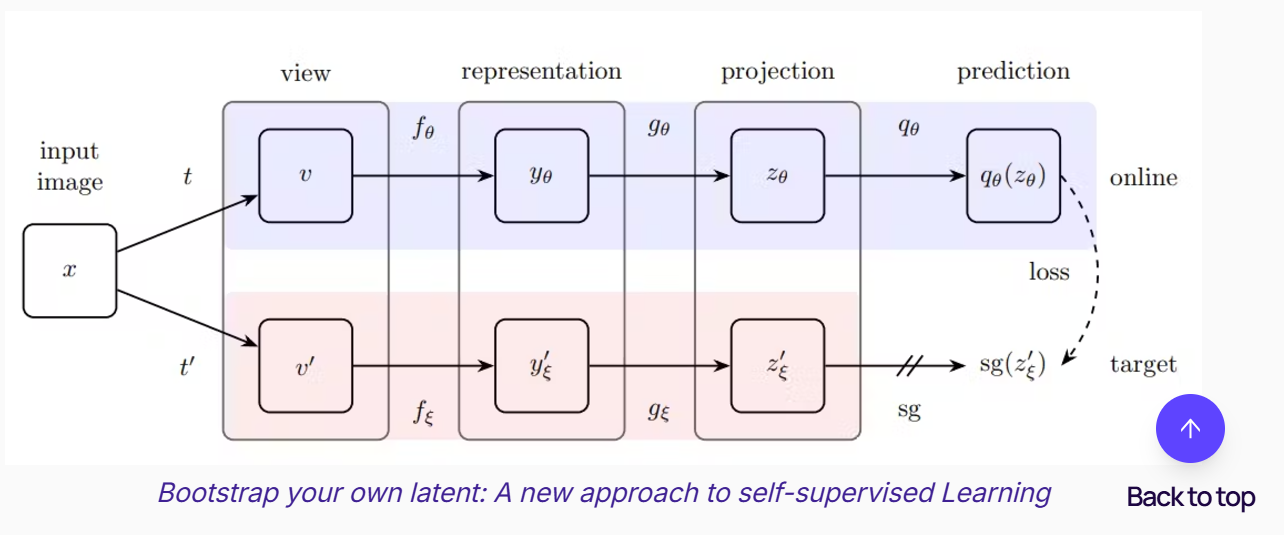
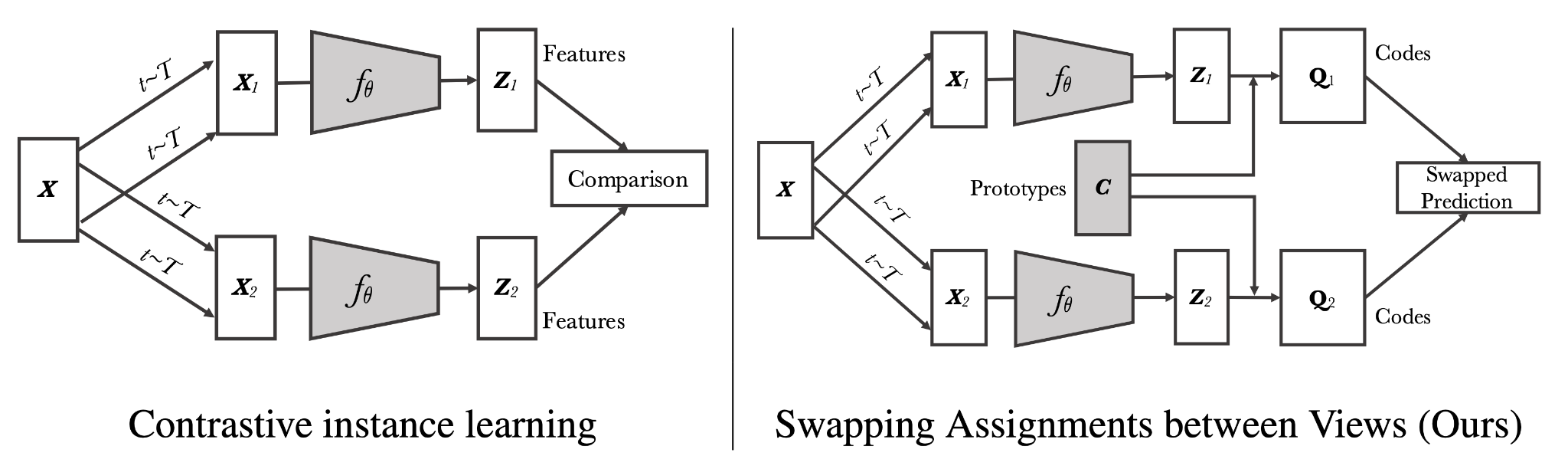
* Spent most of the time doing my presentation
* Contrastive Learning
  + NCE (Noise Contrastive Estimation)
    - Use a logistic regression to find the difference between the “target data” and the “noise”
    - This form is effective with one positive sample and one noise sample
      * Some of the add ons to this algorithm have more than one negative sample
    - 
  + InfoNCE
    - “Categorical cross-entropy loss”
      * Find the positive sample inside a set of noise samples
      * 
    - In sequential prediction, we can use a density function
      * 

11/26/2024

* More on how to implement Contrastive Learning
  + <https://lilianweng.github.io/posts/2021-05-31-contrastive/>
    - Contrastive Learning:
      * “Embedding Space”
      * Things that are alike are near in the space of representations and things that are farther away in the representation space are not alike.
      * Unsupervised and supervised techniques can be used
      * Contrastive learning is very powerful when used in self-supervised learning
    - Contrastive Training Objectives
      * In the start, loss functions only involved one positive and one negative sample
      * Now, there are “multiple positive and negative pairs in one batch.”
      * This increases the accuracy and reliability of the model
      * Contrastive Loss
        + “one of the earliest training objectives used for deep metric learning in a contrastive fashion.”
        + 
      * Triplet Loss
        + 
        + One Anchor (call this “X”), choose a positive sample (X+) and a negative sample (X-)
        + X+ is from the same class as X
        + X- is from some other class
        + Triplet Loss is able to “minimize” how far the X (the anchor) and X+ are and “maximize” how far the X (the anchor) and X- are
        + 
        + Need to have “challenging” X- if we want a good model
      * Lifted Structured Loss
        + “utilizes all the pairwise edges within one training batch for better computational efficiency.”
        + 
        + 
      * N-pair loss
        + The N-Pair Loss is as follows

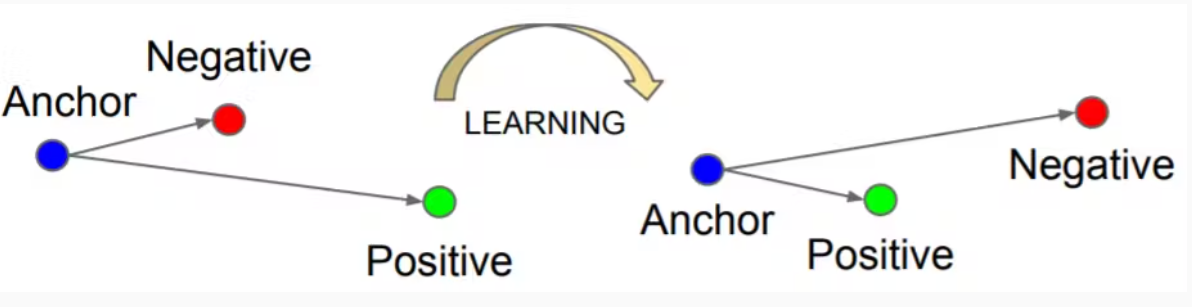


11/21/2024

* <https://encord.com/blog/guide-to-contrastive-learning/#:~:text=For%20example%2C%20in%20sentence%20similarity,similarity%20between%20pairs%20of%20sentences>
  + MoCo
    - Momentum Contrast
    - “self-supervised contrastive learning framework”
    - “It introduces the concept of a dynamic dictionary of negative instances,”
    - This approach uses a momentum encoder, “which gradually updates the representations of negative examples to enhance the model's ability to capture relevant information.”
    - “The framework maximizes agreement between positive pairs while minimizing agreement between negative pairs. ”
    - Works well for NLP
  + BYOL
    - Bootstrap Your Own Latent
    - Updating the “target network parameters”
    - “Pair of online and target networks”
    - “, with the target network updated through exponential moving averages of the online network's weights. BYOL focuses on learning representations without the need for negative examples.”
    - 
    - The agreement is high between “augmented views of the same instance” while “decoupling the similarity estimation from negative examples.”
    - This also could be used for NLP
  + SwAV
    - “clustering-based objectives to learn representations”
    - Multiple different views of the same thing
    - Used mostly in computer vision when trying to identify objects
  + Barlow Twins
    - “reduces cross-correlation between latent representations.”
    - Decorrelation Loss that pushes the model in the direction of making “ diverse representations for similar instances, enhancing the overall discriminative power.”
    - By making cross correlation decrease, this approach is able to find littler and more minute differences in the data. This works to create a better application as a whole
    - This also seems to have promising results in the area of Natural Language Processing
  + Important Takeaways of Contrastive Learning
    - “Learning meaningful representations” from “unlabeled data”
    - Similarity and dissimilarity is used to “map instances in a latent space”
    - Supervised Contrastive Learning with explicit labels
    - Self Supervised Contrastive Learning with “pretext tasks for unlabeled data.”
    - Important parts of Contrastive Learning
      * Data Augmentation
      * Encoder and Projector Networks
      * Finding the important features and likeness among the data
      * Loss functions
        + Contrastive Loss
        + Triplet Loss
        + N-pair loss
        + InfoNCE Loss
        + Logistic Loss
      * Can be used in NLP
* 

11/19/2024

<https://encord.com/blog/guide-to-contrastive-learning/#:~:text=For%20example%2C%20in%20sentence%20similarity,similarity%20between%20pairs%20of%20sentences>

* Triplet Loss
  + Another one of them more popularly used Contrastive Learning Functions
  + The goal is to “preserve the relative distances between instances in the learned representation space. ”
  + Triplets of instances
    - An anchor instance
    - A positive sample (something that is alike the anchor)
      * “Similar to the anchor”
    - A negative sample (something that isn’t alike to the anchor)
      * “Dissimilar to the anchor”
    - Objective
      * Distance connecting the positive sample to the anchor is less than the distance connecting the negative sample and the anchor by a “specified margin”
      * 
    - “The intuition behind triplet loss is to create a "triplet constraint" where the anchor instance is pulled closer to the positive sample while being pushed away from the negative sample. ”
    - The model is able to tell the difference when comparing similar and not similar instances by learning representations that agree with the triple constraint
    - This is a good loss function for some CV tasks, as it helps tell the difference between things where the margin of error is very small
    - The challenge is that it's hard to find good triplets from a really big dataset that will help the model and takes computational resources
* N-Pair Loss
  + “An extension of triplet loss”
  + One anchor is accompanied with more than 1 positive and negative samples
  + With Triplet Loss one anchor is compared to 1 negative and 1 positive sample, here there is more than 1.
  + The goal of N-Pair loss is to “ maximize the similarity between the anchor and all positive instances while minimizing the similarity between the anchor and all negative instances.”
  + N-Pair Loss makes the model figure out subtle relationships within instances, making for “more robust supervision during learning”
  + Because it looks at multiple instances at the same time, N-pair loss finds hidden and hard to decipher patterns and makes the discriminatory power even better in the representations that it produces
  + This is also used when small fine grain things make a lot of difference, requires a lot of computing power especially with big datasets, but can be better than triplet loss because it looks at multiple instances
* InfoNCE
  + “derived from the framework of noise contrastive estimation.”
  + “ It measures the similarity between positive and negative pairs in the learned embedding space. ”
  + “maximizes the agreement between positive pairs”
  + “minimizes the agreement between negative pairs.”
  + Looking at this problem like a binary classifier (assigning a particular thing to one of two attributes)
  + “. Given a positive pair and a set of negative pairs, the model is trained to discriminate between positive and negative instances. The similarity between instances is measured using a probabilistic approach, such as the softmax function.”
  + Used in self-supervised contrastive learning
    - “ where positive pairs are created from augmented views of the same instance, and negative pairs are formed using instances from different samples. By optimizing InfoNCE loss, the model learns to capture meaningful similarities and differences in the data points, acquiring powerful representations.”
* Logistic Loss
  + AKA Logistic Regression Loss or Cross-Entropy Loss
  + “Probabilistic loss function”
  + “ Logistic loss models the likelihood of two instances being similar or dissimilar based on their respective representations in the embedding space.”
  + It makes an educated guess about whether 2 instances are similar or not similar looking at their representations in the embedding space. Being in the same class means they are similar and being in different classes means they are not similar.
  + “By maximizing the likelihood of positive pairs being similar and minimizing the likelihood of negative pairs being similar, logistic loss guides the model toward effective discrimination.”
  + The probabilistic function here also helps when wanting to model complex relationships and get minute differences when comparing instances to each other
  + Can be useful in “image recognition, text classification, and recommendation systems”
* Contrastive Learning use cases:
  + Semi-Supervised learning
    - “Models are trained using labeled and unlabeled data”
    - Labeled data is expensive and hard to make whereas enabled data is very easy to find
    - Semi-Supervised learning is good when you have some labeled and some unlabeled data
  + Supervised learning
    - (used but has disadvantages as well)
  + NLP
    - “For example, in sentence similarity tasks, contrastive learning allows models to learn representations that capture the semantic similarity between pairs of sentences. By leveraging the power of contrastive learning, models can better understand the meaning and context of sentences, facilitating more accurate and meaningful comparisons.”
  + Data Augmentation
    - CV
* Popular Contrastive Learning Frameworks
  + SimCLR
    - Self Supervised
    - “leveraging a combination of data augmentation, a carefully designed contrastive objective, and a symmetric neural network architecture.”
    - “The core idea of SimCLR is to maximize the agreement between augmented views of the same instance while minimizing the agreement between views from different instances. ”
    - “Large-batch training scheme”
    - “"normalized temperature-scaled cross-entropy" (NT-Xent) loss ” (normalization technique)
      * Makes training stability better and the learned representations are of better quality
    - Can be used in NLP
  + MoCo
    - Next class

11/18/2024

<https://encord.com/blog/guide-to-contrastive-learning/#:~:text=For%20example%2C%20in%20sentence%20similarity,similarity%20between%20pairs%20of%20sentences>

* Loss Functions in Contrastive Learning
  + A lot of loss functions can be used to “define the objectives of the learning process.”
  + The loss functions help the model to find meaningful representations and differentiate when looking at similar and non similar instances
* Contrastive Loss
  + “Fundamental loss function”
  + “Maximize the agreement between positive pairs (instances from the same sample) and minimize the agreement between negative pairs (instances from different samples) in the learned embedding space”
  + Similar instances are pulled closer to each other and dissimilar instances are put farther away from each other.
  + Contrastive Loss is usually “defined as a margin-based loss”
    - Similarity when comparing 2 instances is gauged by looking at some sort of distance metric (Euclidean distance or Cosine Similarity)
  + “The contrastive loss is computed by penalizing positive samples for being far apart and negative samples for being too close in the embedding space.”
  + InfoNCE loss:
    - Objective: “widely used variant of contrastive loss”
    - Contrastive loss seems to work well in NLP because it can teach the model how to see differences and similarities which help it discriminate effectively
* Triplet Loss
  + Start from this point next class

11/14/2024

* Contrastive Learning (milestones task 3)
  + <https://encord.com/blog/guide-to-contrastive-learning/#:~:text=For%20example%2C%20in%20sentence%20similarity,similarity%20between%20pairs%20of%20sentences>
    - Contrastive Learning:
      * This happens after the encoding and projecting steps finish
      * Objective:
        + Enlarge the agreement connecting positive pairs

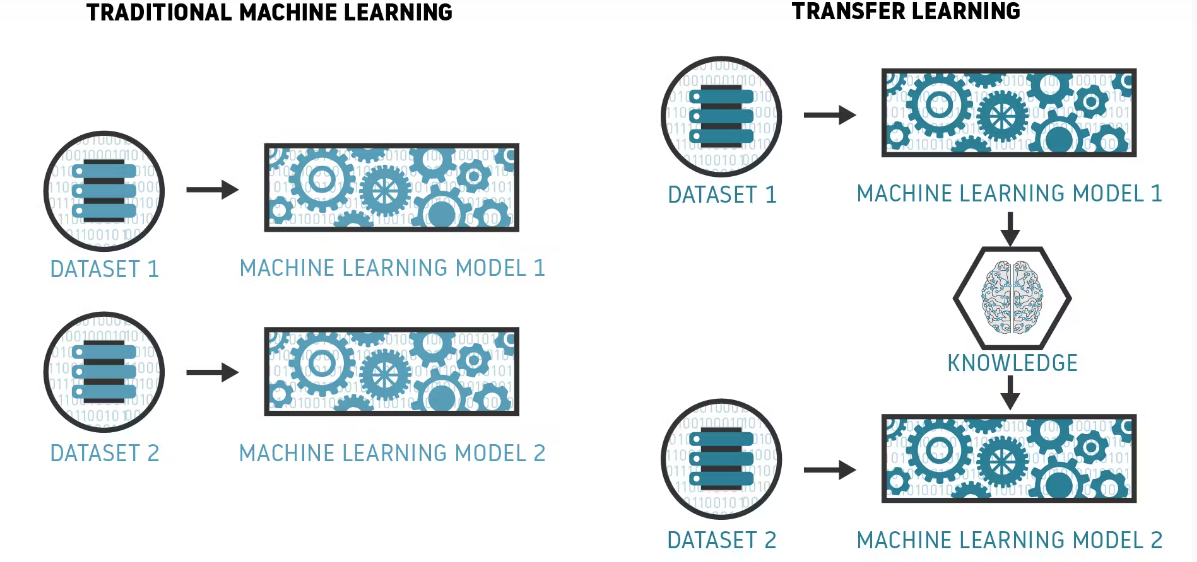
“Instances from the same sample”

* + - * + Reduce the agreement connecting negative pairs

“Instances from different samples”

* + - * + “This encourages the model to pull similar instances closer together while pushing dissimilar instances apart.”
        + How to measure how close instances are: distance metric (ex: Euclidean Distance or Cosine Similarity)
        + Goal: “The model is trained to minimize the distance between positive pairs and maximize the distance between negative pairs in the embedding space.”
    - Loss Function
      * Uses several loss functions to “establish the objectives of the learning process. ”
      * The Loss Functions themselves are very important in leading the model to capture important representations and to be able to see the difference among similar and non similar instances
      * Finding the right loss function to use varies based on the task requirements and data attributes
      * Goal of the loss function: “ Each loss function aims to facilitate learning representations that effectively capture meaningful similarities and differences within the data.”
    - Training and Optimization
      * After you have a loss function made, then the model is trained with a big unlabeled dataset
      * Training Process:
        + “The training process involves iteratively updating the model's parameters to minimize the loss function.”
      * Optimization Algorithms
        + Stochastic gradient descent algorithm

Tweak and make the models hyperparameters really good

* + - * Usually have to use batch-wise updates in the training process:
        + “A subset of augmented instances is processed simultaneously”
      * When training is going on, the model learns how to catch the important features and alikeness inside the data
      * The “iterative optimization process” slowly edits and filters the learned representations, making it so there is more accurate discrimination and division ”between similar and dissimilar instances.”
    - Evaluation and Generalization
      * This is a way of judging how good the learned representations are and how effective they can be in a practical setting
    - Downstream task evaluation
      * The ultimate way of telling whether contrastive learning was successful or not is looking at how well it did on “downstream tasks”
      * The learned representations work as “input features” for “tasks such as image classification, object detection, sentiment analysis, or language translation.”
      * To tell the models performance, we can use metrics like “accuracy, precision, recall, F1 score,” or other things that are specific to each task
      * “ Higher performance on the downstream tasks indicates better generalization and usefulness of the learned representations.”
    - Transfer Learning
      * Contrastive Learning itself leads to Transfer Learning
      * “where the [presentation](https://blog.flipsnack.com/how-to-make-a-presentation/) of learned representations from one task can be applied to related tasks. ”
      * You judge how well the generalizations are when you test how the representations do with different or new datasets or tasks
        + If this is done well, it means the contrastive learning did its job
        + 
    - Comparison with baselines
      * Have to see which is better:
        + State of the art/baseline models OR
        + The learned representations
      * “Comparisons can be made regarding performance metrics, robustness, transfer learning capabilities, or computational efficiency. ”
      * These comparisons allow us to see the benefits of Contrastive Learning and how it might be better than other other approaches
    - Loss Functions (Next class)

11/12/2024

* Contrastive Learning (I need to research more because I plan on using this, milestones task 3)
  + <https://encord.com/blog/guide-to-contrastive-learning/#:~:text=For%20example%2C%20in%20sentence%20similarity,similarity%20between%20pairs%20of%20sentences>.
    - Finding representations that are significant from data that doesn’t have labels on it
    - Looking at similarity and dissimilarity
    - In a space
      * Similar instances are grouped together
      * Dissimilar instances are put farther apart
    - Basics of Contrastive Learning
      * “Contrastive learning is an approach that focuses on extracting meaningful representations by contrasting positive and negative pairs of instances.”
      * Supervised Contrastive Learning
        + The data itself is labeled in order to train models to tell the difference between similar and dissimilar instances
        + The model is trained with data points and labels (telling whether the data points are similar or not)
        + Making a representation space with similar closer and dissimilar farther
        + Information Noise Contrastive Estimation (Info NCE) loss function is a popular way to use

“ maximizes the agreement between positive samples and minimizes the agreement between negative samples in the learned representation space.”

Discrimination with things that are similar and things that are different

* + - * Self-Supervised Contrastive Learning
        + Doesn’t give labels to the data
        + The model learns without labels
        + “SSCL leverages the design of pretext tasks, which create positive and negative pairs from the unlabeled data.”
        + The pretext tasks are made to motivate the model to “capture meaningful features and similarities in the data.”
        + Common Pretext tasks:

Generating augmented views:

“This involves creating multiple augmented versions of the same instance and treating them as positive pairs, while instances from different samples are treated as negative pairs. Training the model to differentiate between these pairs, it learns to capture higher-level semantic information and generalize well to downstream tasks.”

* + - * How does contrastive learning work?
        + “Contrastive learning has proven a powerful technique, allowing models to leverage large amounts of unlabeled data and improve performance even with limited labeled data. ”
        + Similar instanced are fundamentally mapped in near each other in the learning embedding space, instances that are not similar are placed farther apart
        + Steps in Contrastive Learning:

Data Augmentation:

Applying transformations or perturbations to the unlabeled data to make “diverse instances or augmented views.”

This helps the model see different perspectives and more variability

Some of the more used data augmentation techniques are cropping, color transformations, flipping, random crop, and rotation.

Data augmentation helps make sure the model is able to recognize the important information even if there are slight variations in the input data (ie: a rotation)

Encoder Network

“The next step in contrastive learning is training an encoder network.”

“The encoder network takes the augmented instances as input and maps them to a latent representation space, where meaningful features and similarities are captured.”

Encoder network is a deep neural network

Convolutional neural network (image data)

Recurrent neural network (sequential data)

The network itself takes out and encodes “high-level representations” using the augmented instances, helping start the process of discriminating between similar and not similar instances.

Projection Network

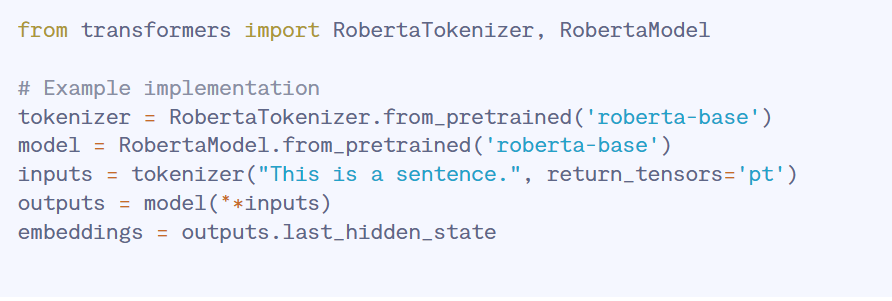
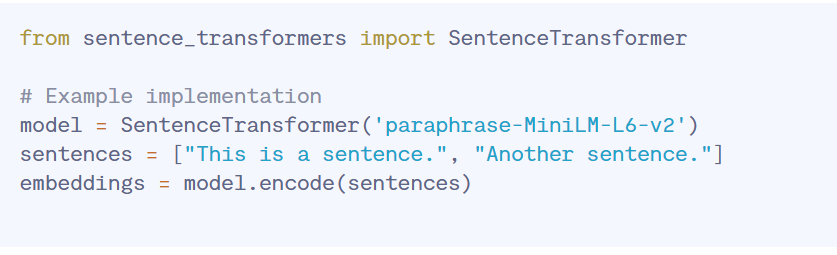
Used to refine the learned representations even more

Input: What the encoder network gave out

Projects that into a “lower-dimensional space”. This space is called the projection or embedding space.

This helps to separate the data into categories of similarity and dissimilarity

11/7/2024

* More on tools for calculating semantic similarity
* <https://www.pingcap.com/article/top-10-tools-for-calculating-semantic-similarity/>
  + RoBERTa
    - Robustly optimized BERT approach
    - A better version of BERT
    - Made by Facebook AI
    - Takes BERT’s model, and makes the training more optimal and ups the amount of training data
    - Really good performance for a Natural language processor
    - Uses fine tooning and takes away some of the training constraints
    - Key Features
      * Optimized training: Makes use of a bigger dataset and training times that are longer
      * Improved performance: Does better on BERT on many different NLP applications
      * Flexibility: Has the ability to be fine tuned for different more specific applications
    - Basic Implementation Steps:
      * Install the Transformers library: Get the transformers library
      * Load the pre-trained model: Use an already trained RoBERTa model
      * Tokenize input text: Using the RobERTa tokenizer, take the text and tokenize it
      * Generate Embeddings: Utilize the model in order to make embeddings for NLP tasks
      * 
    - Example use cases
      * Text Classification: Improve text classification with stronger embeddings
      * Question Answering: Better with understand the users question and potential answers that can be given
      * Semantic Search: Understands the input query much better and that will help users searches
    - Advantages/positives
      * High Performance: Very good results, good on different NLP measuring points and tasks
      * Robust Training: Uses a lot of training optimizations
      * Versatility: Can be used for many NLP tasks
    - Limitations
      * Computationally Expensive: Takes a lot of resources to train it
      * Complexity: Hard to implement and fine tune
  + Sentence-BERT
    - Its a modified version of BERT only for sentence embeddings
    - Has a triple network structure
    - Does really well on semantic similarity and paraphrase detection
    - Key Features
      * Sentence Level Embeddings
        + Makes embeddings for the whole sentences
      * Efficient Similarity Computation
        + Optimized for taking sentence pairs and comparing and contrasting them
      * Pre Trained Models
        + Can be used very quickly as its already trained
    - Basic Implementation Steps
      * Install the Sentence-Transformers Library
        + Make use of the sentence-tranformers library
      * Load the already trained model
        + Pick an already trained model
      * Encode Sentences
        + Make sentence embeddings
      * Utilize Embeddings
        + Use in NLP tasks
      * 
    - Example use cases:
      * Semantic Textual Similarity: Find how similar sentences are for plagiarism detection
      * Paraphrase Detection: Look at sentences that have the same meaning but are differently phrased
      * Text Clustering: Puts similar texts together looking at the semantic content of the text
    - Advantages
      * High Accuracy:
        + Makes really good and useful sentence embeddings
      * Efficiency
        + Its optimized in order to make similarity comparisons really quickly
      * Versatility
        + Can be used for many NLP tasks with sentence comparisons
    - Disadvantages
      * Resource Intensive: In order to train and inference you need a lot of computing power
      * Fixed Context: Cannot really change for different contexts without refraining

11/6/2024

* Another tool for calculating semantic similarity
  + <https://www.pingcap.com/article/top-10-tools-for-calculating-semantic-similarity/>
  + ELMo
    - Embeddings for language models
    - Made by the Allen Institute for AI
    - Makes word representations that are sensitive to the context for which they are in. This is done with a bidirectional LSTM (Long Short-Term Memory) network
    - Goes through the text in both directions (including forward and backward) which helps it make really good semantic embeddings for words
    - Important features:
      * Contextualized embeddings: Makes different embeddings for the same word based on the context it appears in. Values the contextual situation of the word
      * Deep Learning: Uses a deep bidirectional LSTM model
      * Pre trained models: Don’t need to train it much, it's already been trained. This is also time efficient for the user as they don’t have to focus on training the model itself. It can be used as soon as you load it.
    - How to implement it:
      * Install the required libraries: allennlp and torch
      * “Load Pre-trained Model: Download and load the ELMo model.”
      * “Tokenize Text: Prepare your text data for embedding generation.”
      * Generate Embeddings: Make use of the model to make word embeddings that are tailored to the context of the word (taken care of by the model itself)
      * from allennlp.modules.elmo import Elmo, batch\_to\_ids

from allennlp.modules.elmo import Elmo, batch\_to\_ids

# Example implementation

options\_file = "https://allennlp.s3.amazonaws.com/models/elmo/2x4096\_512\_2048cnn\_2xhighway/elmo\_2x4096\_512\_2048cnn\_2xhighway\_options.json"

weight\_file = "https://allennlp.s3.amazonaws.com/models/elmo/2x4096\_512\_2048cnn\_2xhighway/elmo\_2x4096\_512\_2048cnn\_2xhighway\_weights.hdf5"

elmo = Elmo(options\_file, weight\_file, 2, dropout=0)

sentences = [['This', 'is', 'a', 'sentence'], ['Another', 'sentence']]

character\_ids = batch\_to\_ids(sentences)

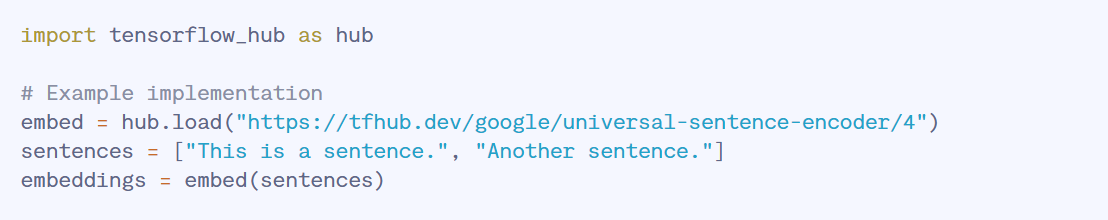
embeddings = elmo(character\_ids)

* Example use cases:
  + Name entity registration: “Improve entity recognition by leveraging context-aware embeddings.”
  + Sentiment Analysis: “Enhance sentiment classification by understanding the nuanced meanings of words.”
  + Machine Translation: “Improve translation quality by capturing the context of words and phrases.”
* Advantages:
  + Is able to decipher context of words
  + Can be used for many NLP processes
  + Model is already trained, so you don’t need to spent time and computer power for training
* Limitations
  + Takes a lot of resources
  + Complex, hard to make

10/31/2024

* Worked on revising my presentation (most of class)
* Research into InferSet: <https://www.pingcap.com/article/top-10-tools-for-calculating-semantic-similarity/>
  + Uses supervised learning
  + Made by the company Facebook
  + Trained with Stanford National Language Inference dataset
  + Good model for understanding semantics between sentences
  + Important Features:
    - Supervised learning: The data itself is labeled in order to make better inferences about relationships between sentences
    - High Quality Embeddings: Makes very good sentence embeddings that does well on many different benchmarks
    - Compatibility: Does good on other machine learning frameworks or models
  + How to make it:
    - Install the libraries torch and InferSent
    - Load the model that has already been trained: Download the InferSent model
    - Encode sentences: Make sentence embeddings with the model that you already loaded and trained
    - Apply the embeddings: Use the embeddings made for any tasks relating to NLP or semantic similarity
  + 
  + Use Cases:
    - Looking at text similarly, for things like checking originally
    - Answering questions better: For the computer to actually process and understand what the user asked it to do
    - Text Summarization: Writing summaries by getting the important parts of the document
  + Advantages
    - Really good performance
    - Supervised training: Uses the labeled data to its advantage to make really effective embeddings
    - Versatile: Can be used for a variety of purposes, one of them being semantic similarity
  + Limitations:
    - Need good training data for good performance
      * If the training data isn't good, then the performance will suffer
    - Resource intensive: Need a lot of computational power to training and use

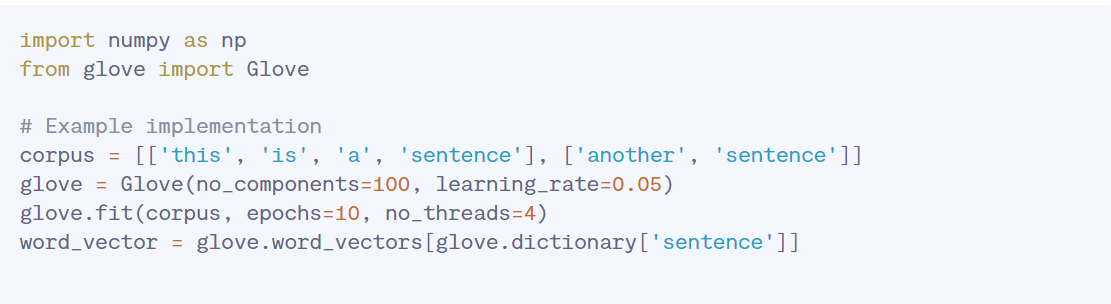
10/29/2024

* First part of class: Research for project
  + <https://www.pingcap.com/article/top-10-tools-for-calculating-semantic-similarity/> (more research into calculating semantic similarity scores, milestone 2)
    - Universal Sentence Encoder
      * Take sentences and makes them high-dimensional vectors that is able to recognize semantic similarity
      * Is centered on sentence level embeddings
      * Can be used when one desires to understand what a bunch of sentences or paragraphs mean
      * Encoding sentences, to get the meaning out of it
      * How to implement:
        + Use tensorflow library in Python
        + “ Load the Universal Sentence Encoder from TensorFlow Hub.”
        + Encode each sentence
        + Use the embeddings in whatever NLP tasks you need
        + 
      * Use cases:
        + Semantic Search
        + Text Clustering: Puts alike texts next to each other looking at semantic content
        + Paraphrase detection: Look at sentences that actually have the same meaning with the sentence embeddings
      * Advantages
        + Effectively gets the meaning of sentences
        + Not too hard to use
        + A lot of different use cases in the real world
      * Limitations
        + Takes a lot of computational power
        + The content is fixed: “Cannot dynamically adapt to new contexts without retraining.”
* Second part of class: Corrections for presentation

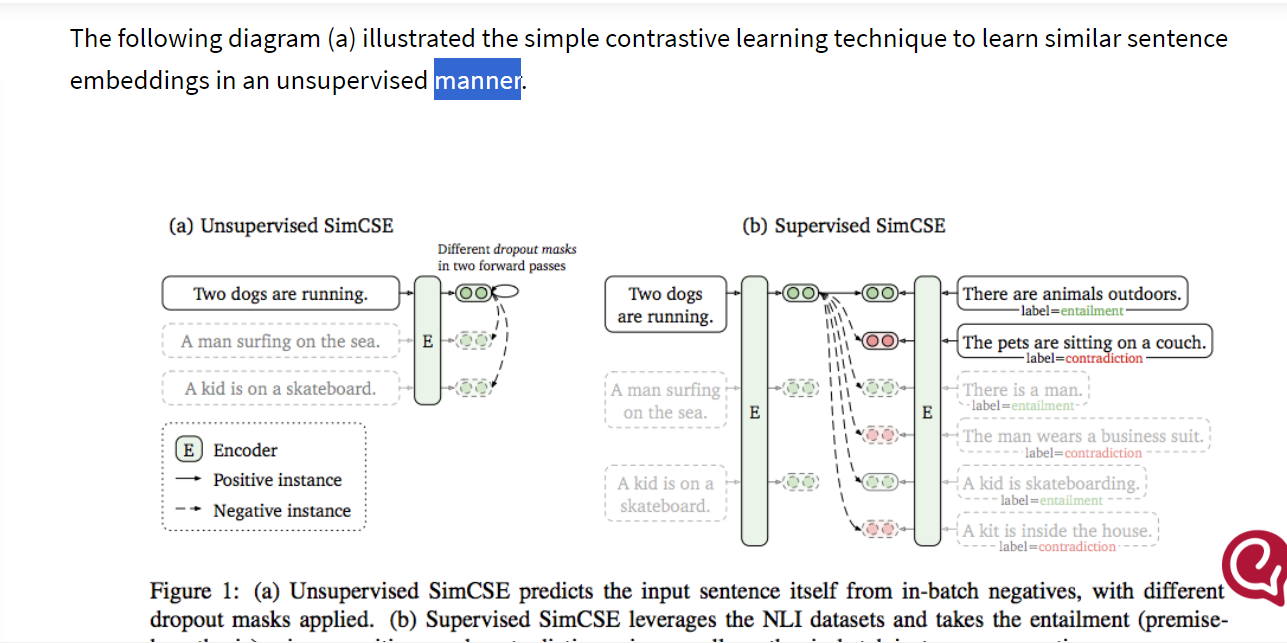
10/25/2024 (I will be absent on Monday 10/28 so I am doing this journal entry early)

* <https://www.pingcap.com/article/top-10-tools-for-calculating-semantic-similarity/> (more research into calculating semantic similarity scores, milestone 2)
  + BERT
    - Looks at text bidirectionally (goes from both directions)
    - Looks at left and right context
    - Uses pre-trained models for NLP
    - Has the ability to be fine tuned for an application
    - Steps to use BERT:
      * Take a library (like Transformers), which gives an already trained model
      * Tokenize whatever is coming in with the BERT’s tokenizer
      * Fine tune the model for things that you want it to do specifically
      * 
      * Use cases: Semantic similarity search
    - Advantages: Very accurate, can do many different NLP tasks, has good contextual understanding (semantic relationships)
    - Disadvantages: complex and takes time to implement and find tune, takes a lot of computational memory
  + FastText
    - Improvement upon Word2Vec for words not in the vocabulary
    - Character n-grams to make word vectors
    - Can train on big datasets fast
    - Makes use of “hierarchical softmax function” to calculate probabilities faster
    - Pre-trained models can be used
    - Implementation steps:
      * Install fasttext library
      * Tokenize and normalize data as part of the preprocessing
      * Train the model with the train\_unsupervised function
      * “Embedding Extraction: Extract word vectors for downstream tasks.”
      * 
    - Some use cases:
      * Classifying text
      * A model that can deal with complex words
    - Advantages:
      * Really good with outside of the vocabulary words
      * Its fast
    - Disadvantages
      * Doesn’t see context as well as BERT did
      * Isn’t as adaptable to new contexts

10/24/2024

* <https://www.pingcap.com/article/top-10-tools-for-calculating-semantic-similarity/>
  + Third article for milestone 2
  + Semantic similarity: How close two pieces of text are based on the meaning of the text
  + Helps the machine understand human language better
  + Not only looking at the words themselves, but also looking the context and concepts within the text
  + Usually uses (to get the meaning of the text):
    - Word embeddings: “representation of the word”. “Typically, the representation is a real-valued vector that encodes the meaning of the word in such a way that the words that are closer in the vector space are expected to be similar in meaning.”
      * <https://en.wikipedia.org/wiki/Word_embedding>
    - Vector space models: “Vector space model or term vector model is an algebraic model for representing text documents as vectors such that the distance between vectors represents the relevance between the documents. It is used in information filtering, information retrieval, indexing and relevancy rankings.”
      * https://en.wikipedia.org/wiki/Vector\_space\_model
    - Deep learning techniques: Using strategies that teach a computer to learn like human brains.
  + Best ways to calculate semantic similarity
    - Word2Vec
      * Words > continuous vector values (it is able to get the semantic relationships between words)
      * 2 layer neural network
      * 2 architectures:
        + Continuous Bag of Words
        + Continuous skip-gram
      * Efficient to train
      * CBOW: predicting the target word using words around it
      * Skip-gram: Predict words around a target word
      * Looks at semantic and syntactic relationships within words
      * How to implement:
        + Get and preprocess data
        + Choose one of the two models
        + Train model with a library such as gensim
        + Extract word vectors for more data analysis
        + 
      * Limitations: Not good with words that have more than 1 meaning, only has a limited vocabulary
    - GloVe
      * Word Context matrix, factorization process
      * Uses matrix factorization to make the word vectors
      * Semantic and syntactic analysis on the words
      * How to implement it:
        + Find a big text corpus
        + Construct a matrix of word co-occurrences
        + Use factorization techniques to get the word vectors
        + “Use the generated embeddings in downstream tasks.”
        + 

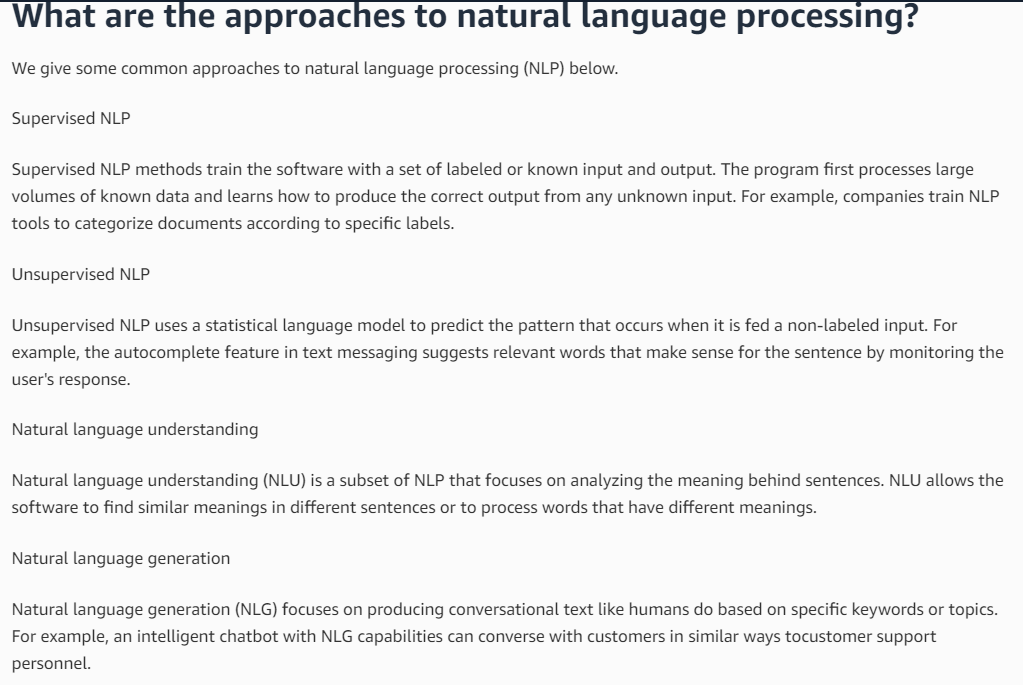
10/22/2024

* <https://www.engati.com/blog/contrastive-learning-in-nlp> (contrastive learning in NLP)
  + Establishing similarities and differences between samples, grouping together samples
  + Has done well for language tasks
  + How does contrastive learning in NLP work?
    - Contrastive definition: “showing the difference between two things when you compare them”
    - Negative samples are put further away and positive samples are placed closer in the vector space
    - This method can be applied to supervised or unsupervised learning
      * Seems to do really well on semantic similarity
    - Might be able to do this without labeling the datasets, can be used for semantic similarity
    - Unsupervised learning is easier as you don’t have to go through the process of labeling the data
      * Unsupervised models don’t have an idea about what the right answer could be
    - Take the input and give it to a pre-trained encoder and save this in vector space
    - 
  + “Hence, for better representation of the process and data, we use contrastive learning with self-supervised learning.”
  + 
* <https://www.v7labs.com/blog/contrastive-learning-guide>
  + Talks about using a contrastive learning approach
  + Can be used to make the retrieval methods better
    - “positive pairs are paraphrased or otherwise noised sequences paired with their generations, while negative pairs are humans written continuations paired with the machine-generated text”
* [https://aws.amazon.com/what-is/nlp/#:~:text=Natural%20language%20processin](https://aws.amazon.com/what-is/nlp/#:~:text=Natural%20language%20processing%20(NLP)%20is,manipulate%2C%20and%20comprehend%20human%20language)

[g%20(NLP)%20is,manipulate%2C%20and%20comprehend%20human%20language](https://aws.amazon.com/what-is/nlp/#:~:text=Natural%20language%20processing%20(NLP)%20is,manipulate%2C%20and%20comprehend%20human%20language).

* + - Real time comprehension of human written text
    - Analysis and response to human input and the humans sentiment
    - It can differentiate the little things in human speech: dialects, slang, grammar, and other nuances
    - Uses “computational linguistics, machine learning, and [deep learning](https://aws.amazon.com/what-is/deep-learning/) models to process human language.”
    - Computational linguistics:
      * Making a model to comprehend human language
      * Computers use syntactic and semantic analysis to understand humans language
    - Machine learning:
      * Training a model with data to improve its accuracy and efficiency
      * Machine learning helps models understand features of human language including sarcasm, grammar, etc
    - Deep learning:
      * Field within machine learning
      * Works like humans think, computers are thinking like humans do
      * Neural network that has nodes that are similar to the human brain itself, they are able to find patterns in the data and learn them
    - NLP implementation
      * Step 1: Take some unstructured text
      * Step 2: Preprocessing: “The NLP software uses pre-processing

techniques such as tokenization, stemming, lemmatization, and stop word removal to prepare the data for various applications. ”

* + - * + “Here's a description of these techniques:
* Tokenization breaks a sentence into individual units of words or phrases.
* Stemming and lemmatization simplify words into their root form. For example, these processes turn "starting" into "start."
* Stop word removal ensures that words that do not add significant meaning to a sentence, such as "for" and "with," are removed. ”
  + - * Step 3: Training: “Researchers use the pre-processed data and machine learning to train NLP models to perform specific applications based on the provided textual information. Training NLP algorithms requires feeding the software with large data samples to increase the algorithms' accuracy. ”
      * Step 4: Deployment and inference
        + Take an input and predict the output, record accuracy
    - NLP Tasks
      * Take human text and cut it into smaller chunks that computers can comprehend
      * 

10/21/2024 CATCH UP DAY

* Vectorization/making meaning out of words (working on milestones doc Task 2):
  + Article 1: <https://www.enjoyalgorithms.com/blog/word-vector-encoding-in-nlp>
    - Bag of words:
      * Bags are put and defined inside a corpus
      * Used to classify documents and find how many times words appeared in the document
      * Two Steps:
        + Making a vocabulary, not caring about grammar
        + Finding how many times a vocabulary word happened in a certain context
      * In this approach, the actual order and structure of words is not relevant
      * This approaches cares about the words in a document and how often they occur
        + You tend to think that documents that are similar have similar words within them
      * Pros:
        + Intuitive and easy to understand/implement
        + Gives good results when doing doc classifications
      * Cons
        + Doesn’t give info on the order of the words or the structure, which is critical in thinking about NLP
        + When there are a lot of new words, the time to compute goes up. Making it harder for the code to run and taking longer time
  + Possible improvements upon retrieval:
    - Use a different retrieval encoder instead of BM25?
    - Change the threshold
    - Response length
    - Restrict retrieval to a fixed time period
    - Choose a better retrieval strategy
      * Beir
      * “Fine-tuning dense retrievers for the detection task. The retrievers in our work are not fine tuned for the task of AI-generated text detection. However, we hypothesize that fine-tuning retrievers on this task can help retrievers adapt better to the retrieval corpus and detection task. Specifically, a contrastive learning approach could be adopted here: positive pairs are paraphrased or otherwise noised sequences paired with their generations, while negative pairs are humans written continuations paired with the machine-generated text.”
  + Article 2: [https://aws.amazon.com/what-is/nlp/#:~:text=Natural%20language%20processin](https://aws.amazon.com/what-is/nlp/#:~:text=Natural%20language%20processing%20(NLP)%20is,manipulate%2C%20and%20comprehend%20human%20language)
    - [g%20(NLP)%20is,manipulate%2C%20and%20comprehend%20human%20language](https://aws.amazon.com/what-is/nlp/#:~:text=Natural%20language%20processing%20(NLP)%20is,manipulate%2C%20and%20comprehend%20human%20language).
      * Will look into this as article 2 next class

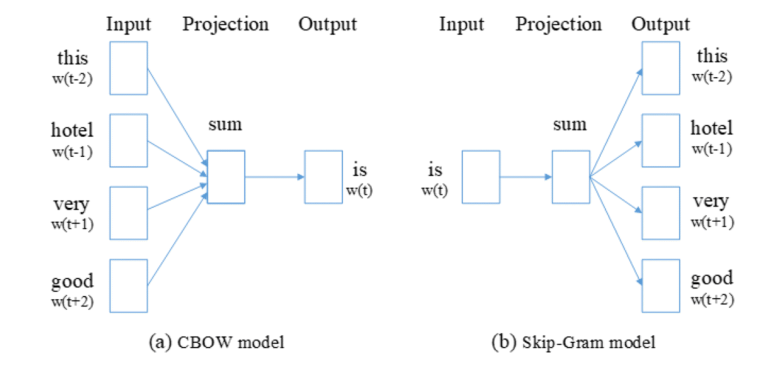
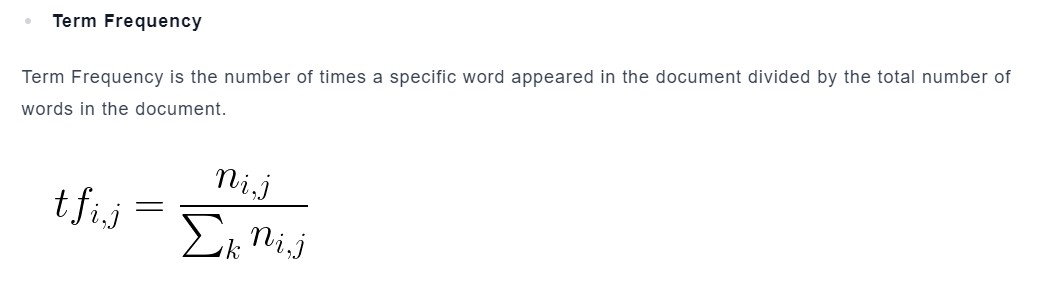
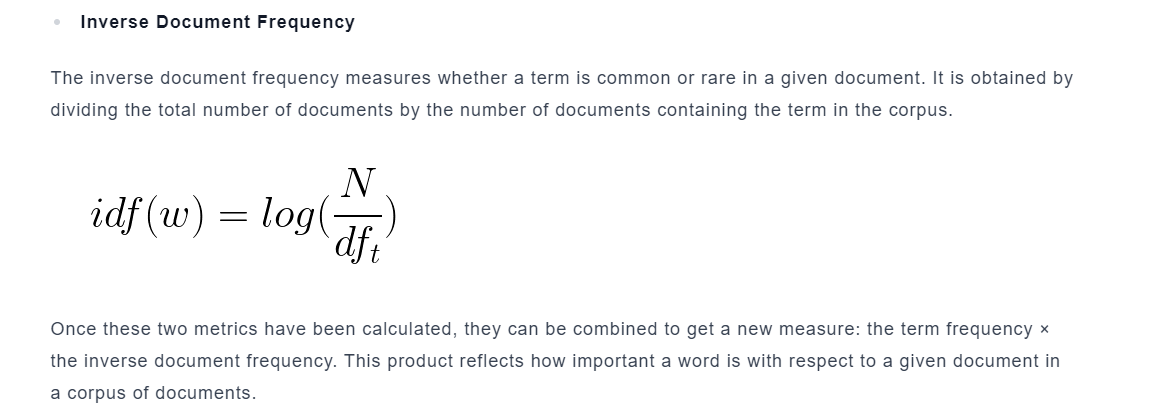
10/17/2024

* Listened to presentations and gave feedback

10/15/2024

* Listened to presentations and gave feedback

10/12/2024

* Word2Vec
  + 2 layer neural network
  + Input: corpus, output: “set of feature vectors representing the words in that corpus”’
  + 2 method in this model
    - Continuous Bag-of-Words Model (CBOW)
      * Can predict the middle word based on words that are around it, the words around it are a few words before and a few words after. Order of the words is not important
    - Continuous Skip-gram Model
      * Predicts some words before and after the current words in a certain sentence
    - 
  + Word2Vec is akin to an autoencoder, encoding every word in to a vector
    - Trains words versus their closer words
    - Go from context to a target word (CBOW) or word to a context (skip-gram). Skip gram is seen as better on larger datasets
  + Word2Vec is a bit difficult to train and doesn’t explain the context
* TF-IDF
  + Term Frequency and Inverse Document Frequency, looking at how important a word is in a document
  + Weighting each word
  + “And” for example doesn’t have much weight
  + 
  + 
  + 
  + Cons: Results might not be the test and takes high memory
* <https://neptune.ai/blog/vectorization-techniques-in-nlp-guide> (another link for vectorization)

10/10/2024

* Research into retrieval methods (looking for some inspiration for my project idea)
  + <https://arxiv.org/pdf/2312.10997>
    - LLM have some limitations, including giving incorrect information, knowledge that is from the past, and unclear reasoning
    - The problem lies in specific tasks, where LLM’s might not have been trained well
      * Creating hallucinations (false information) beyond the information they have access to
      * Retrieval Augmented Generation can help AI’s by taking important parts of documents and adding that information to the AI’s response
    - Overview of RAG
      * People can ask ChatGPT or other AI’s about recent news, where AI doesn’t know the answer
      * RAG helps this by giving information from outside databases
        + In this example, the database can give the AI relevant information about the recent news to enhance the AI’s response
        + This gives the AI to power to give reliable and relevant information as a response
        + Naive RAG

Indexing, retrieval, and generation (known as the “Retrieve-Read framework)

Indexing: clean and extra data from various formats including HTML, Word, PDF, and Markdown, this is then changed to a “uniform plain text format”

Text is then divided into smaller chunks

Then those chunks are taken and encoded into vector representations with an embedding model and put inside a vector database

Retrieval: After taking in a user input, the RAG system uses the same process that was used to encode the database on the input. The input query is made into a vector representation.

Calculates the similarity score between the query vector and the “vector of chunks

<https://www.enjoyalgorithms.com/blog/word-vector-encoding-in-nlp>

Talks about how to make vectors in computer science

Take the text and makes a meaning out of it in terms of numbers so that the computer can understand it

Used in a lot of natural language processing tasks in general

Categorize words together by similarity

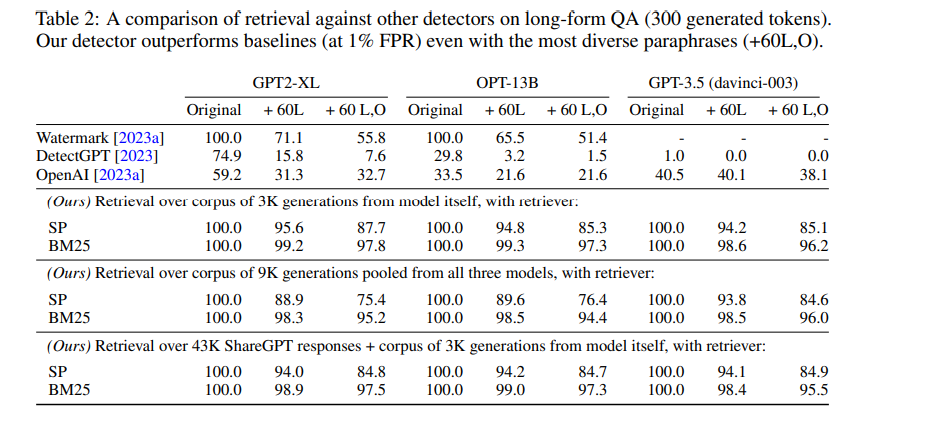
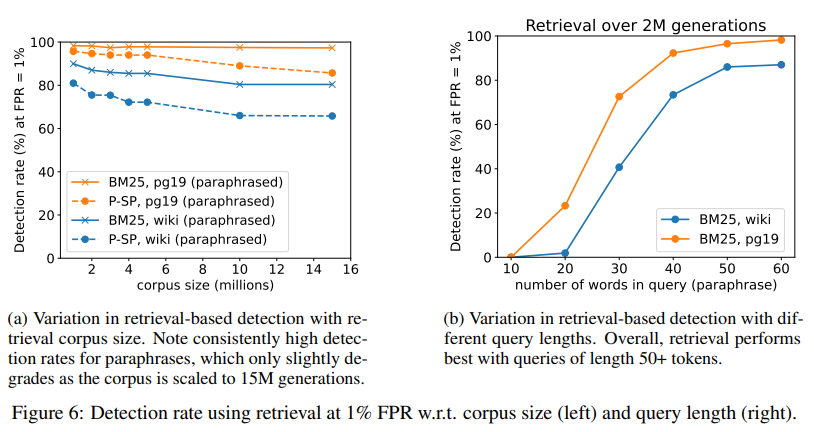
Word2Vec seems like a popular way people make vectors in NLP

* Listened to presentations rest of class

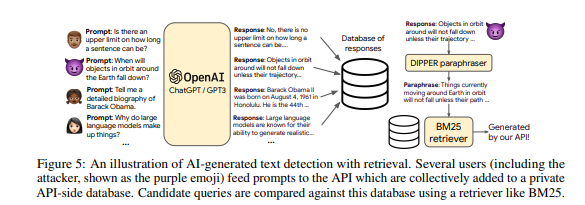
10/8/2024

* Provided feedback to the presentations

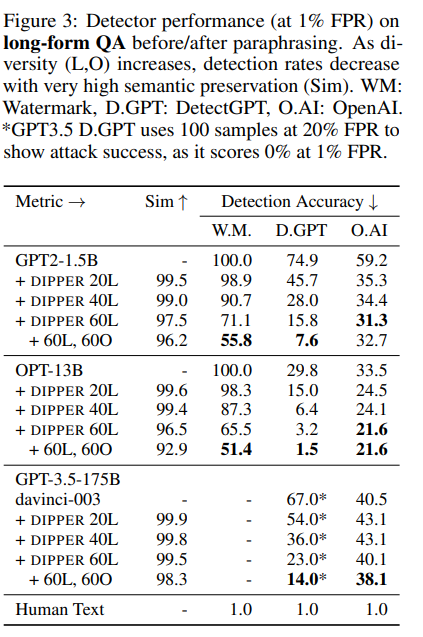
10/7/2024

* “5.2 Controlled comparisons of retrieval with other AI-generated text detectors”
  + They are comparing current AI detectors to the retrieval method “on long-form question answering”
  + 3 databases
    - “1. 3K sequences generated by a specific LM for one of the tasks;”
    - 
    - “2. 9K sequences formed by concatenating the generations from all three” LMs in this paper;
    - “3. 46K sequences constructed by combining the 3K sequences from (1) with 43K LLM responses from ShareGPT-Vicuna”
    - Number 2 is a more difficult test because the retriever has to tell the difference between different generations made from different LLM models with the same input prompt
    - Number 3 is more practical in the real world because it uses many outputs from different LLMs generated from ShareGPT.
    - They do retrieval using “different types of queries”
      * Primary AI generated text
      * Paraphrased text using DIPPER
      * Human written text
      * Each of these has at least 50 tokens
  + Takeaways:
    - Retrieval works better than regular AI detectors
    - BM25 works well on text that is paraphrased, getting a 97.8% on text that has the highest diversity paraphrasing (L60, O60) on GPT2XL, 97.3% on OPT-13B, and 96.2% on GPT3.5 on longer answers of text. This is much improved from previous detector accuracies (55.8%, 51.4%, 38.1%).
    - “Even on our harder augmented databases, detection rates continue to be high: 95.2%, 94.4%, 96.0% for the 9K augmented database; 97.5%, 97.3%, 95.5% for the ShareGPT augmented database. Finally, we observe that BM25 is a more effective retriever than P-SP, scoring 95.2% vs 75.4% on the augmented setting in GPT2-XL. These trends are consistent across different FPR thresholds, as shown in Figure 4”
    - There is also some introductory encouraging results on the potential use of retrieval against text mixing cheating attacks
* “5.3 Is retrieval an effective detector with a large retrieval corpus?”
  + Wonder how does retrieval work with millions and millions of text generations
  + To test the scalability, they take a large amount of AI generated text (over 70 million AI generated sequences). They use the training data that trained RankGen
  + “We use the Project Gutenberg and Wikipedia splits of the training data, each of which contain 15M sequences generated by a T5-XXL model [Raffel et al., 2020] fine-tuned on the different documents in the same domain. We discard generations which are shorter than 50 tokens, and paraphrase a subset of 2K generations to evaluate retrieval. ”
  + “Retrieval is effective even with a corpus size of 15M generations.”
    - 
    - The accuracy of detecting AI text remains pretty high even with different corpus sizes (going from 1M up to 15M generations)
    - There is slight decline as the corpus size increases, 1 percent drop for Project Gutenberg and 9.6 percent for Wikipedia.
    - BM25 remains better than P-SP with different corpus lengths
  + “Retrieval detection works best with 50 or more tokens of generated text.”
    - Queries that are of a shorter length are more probable to get more matches (some of these being fake) versus longer ones
    - “We observe that BM25 struggles to detect paraphrased text with a query length of 20 (less than 25% accuracy), but the detection rate rapidly increases and begins to plateau at 50 tokens.”
* “5.4 Scalability and limitations of retrieval-based detectors”
  + This person believes that retrieval is a scalable approach
  + Important limitations
    - 1: “potential privacy risk of exposing all LLM responses behind a binary classifier; ”
    - 2: “inability to use retrieval-based detection on open-source LLMs like LLAMA ”
    - 3: “the need to implement and maintain retrieval infrastructure.”
* “Appendix”
  + “A Limitations of retrieval-based detection and ideas for scaling it further”

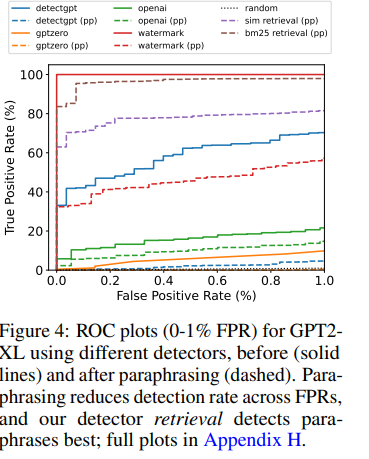
10/6/2024

* Answering the question of how a detector can defend against paraphrasing of AI
* Using “retrieval over previously-generated sequences as a defense against paraphrase attacks.”
* Higher level of what's going on: “an API provider first stores every sequence generated by their LLM in a database. The API provider offers an interface that allows users to enter candidate AI-generated text as a query. The interface searches over the entire database of previously-generated text, trying to find a sequence that approximately matches the content of the input query. This search can be done using a semantic similarity scorer like P-SP [Wieting et al., 2022] or a retriever like BM25 [Robertson et al., 1995]”
* It should be able to match sentences to their original meaning because semantically the paraphrasing still has a similar meaning to the original text
* 
* “5.1 Formulating the retrieval defense”
  + Building the database:
    - Have prompts fed into the LLM API (ex: GPT 3.5) considered x and those have outputs y. They make a database that stores every encoded LLM API output using retrieval encoder, this is storeid on the API side. Clients cannot see it unless they go by the API in the next step
  + Querying the database:
    - The way we can check if text was indeed paraphrased and generated by AI is looking at candidate text and testing its encoded vector against an entry in the database and see if the similarity score is higher than some threshold, “chosen by the API provider”
    - 
    - “We expect unperturbed machine-generated text to always get a score of 1.0, while paraphrasing the text may lower the detection score. Hence, lowering T will increase the detection rate of heavily paraphrased text but also increase the false positive rate (i.e., human-written text that resembles sequences previously generated by the LLM API can be falsely flagged). Since N can be very large, the score can also be approximated using efficient nearest neighbor libraries like FAISS [Johnson et al., 2019]. However, in this work we only compute exact inner products. As the retriever fret, we experiment with two choices: P-SP [Wieting et al., 2022] and BM25 [Robertson et al., 1995]. We implement BM25 using the retriv library from Bassani [2022]. In order to normalize and calibrate BM25 scores, we compute the F1-score unigram token overlap [Rajpurkar et al., 2016] between the candidate y ′ and the best retrieval y∗ to get a detection score in [0, 1].”

10/1/2024

* <https://arxiv.org/pdf/2303.13408>
  + “4.2 Models, datasets & detection algorithms”
    - “Basic Language Models”: GPT2-XL model, OPT-13B model, and the “text-davinci-003 variant from the GPT-3.5family”
    - For all the language models, they use a text generation that is 300 tokens big and then run that text through dipper in order to paraphrase it
    - “Natural language generation tasks”:
      * First, Open ended generation: When the language model generates an extension of the two sentence prompt
        + they get the prompts from WikiText-103 and sample that, the next 300 tokens are considered to be written by humans
      * Second, long form question answering
        + A language models gives a response that is 300 words long
    - “Detection Algorithms”:
      * Using 5 different detectors:
        + Watermarking
        + DetectGPT
        + GPTZero
        + OpenAI’s text classifier
        + RankGenXL-all
      * Default hyperparameters for all of the detectors
      * They make sure to adhere to the minimum length requirements, they removed any text smaller than the minimum length
    - “Paraphrasing AI-generated text”:
      * Dipper gets the prompt and the AI generated answer to the prompt as an input
      * The input is as follows: “lexical = L, order = O prompt <p> generated-text </p>”
      * They make L = 20, 40, 60 and O = 0, 60 in the experiments
      * They also make sure that the AI generation, the paraphrased text, and human written text are the same length. If they aren’t, then they truncate to whichever of the 3 had the shortest length
      * They iteratively paraphrase longer pieces of text, 3 sentences at a go.
      * They only paraphrase the text once, they don’t do it multiple times
      * 
  + “Attacking AI-generated text detectors”
    - Key Takeaways:
      * “Paraphrasing significantly lowers detection accuracy while preserving input semantics.”
        + As the text is paraphrased, the detection accuracy goes down for all LM’s, tasks, and detectors.

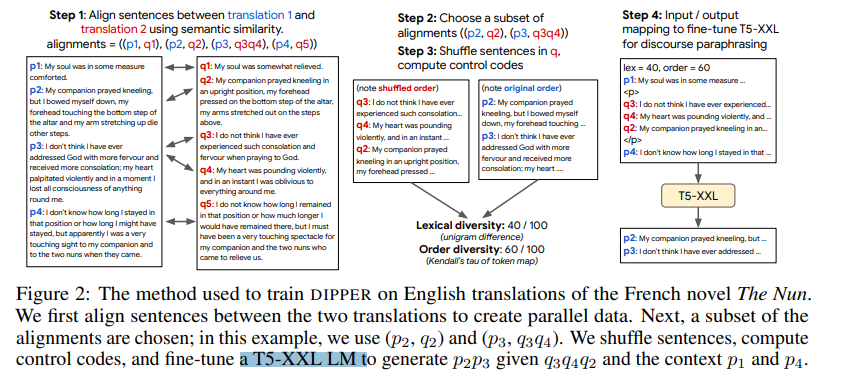
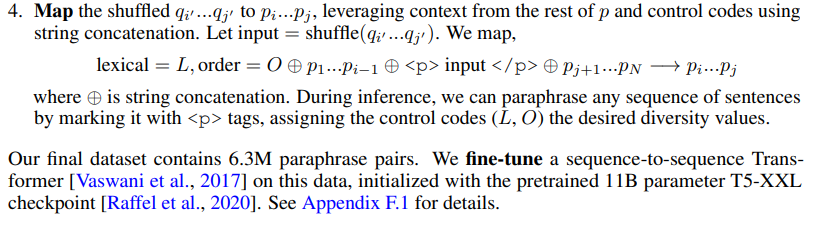
DetectGPT goes from 70.3 % to 4.6 %

* + - * + Dipper keeps the semantics very similar, 88-99 percent of paraphrases have a P-SP score larger than the median score of human made paraphrases
        + Watermarking is the best detection that resists the paraphrased text the most
      * “Non-watermarking detectors are generally ineffective”
        + These detector generally don’t do well against text made from larger models such as OPT-13B and GPT-3.5, with detection accuracies less than 50 percent
        + DetectGPT works well on the GPT2-XL model (around 75 percent on the longer question and answer), the accuracy plummets to 29.8 percent on OPT-13B
        + GPTZero and RankGen aren’t good at all, can detect less than 15 percent AI generated text that isn’t even paraphrased
      * “ROC plots confirm the trends at different false positive rates.”
        + No matter the false positive rate, paraphrasing makes the detention percentage decrease significantly
        + 
  + “4.4 Alternative paraphrasing attacks“
    - “Paraphrasing multiple times”
      * They paraphrase one time in their experiments but it's possible to paraphrase multiple times. An add-on can be to sample more than one time from DIPPER and “ choose a paraphrase that evades the detector while also preserving semantics.”
      * This assumes that the plagiarizer/AI user has access to a detector (not necessarily always true)
    - “Non-DIPPER paraphrasers”
      * Can use paraphrasing machines that aren’t DIPPER, they use sentence level paraphrasing
      * Go sentence by sentence, not paying attention to the prompt at hand. Usually used for larger pieces of text.
      * Non-DIPPER paraphrases are not as good in terms of quality paraphrasing and aren’t as good with the prompt. They don’t have the details like diversity control and input support that DIPPER has (these are important because a student might want to use both of those properties when paraphrasing text)
      * Can also try to ask ChatGPT itself to paraphrase content and see the result of that
        + Potentially could get caught by watermarking
  + “5 Defense against paraphrase attacks using retrieval”
    - How can a detector defend against paraphrased text?
    - “ In this section, we propose retrieval over previously-generated sequences as a defense against paraphrase attacks. ”
    - PICK UP AT: 5 Defense against paraphrase attacks using retrieval

9/30/2024

* Experiments attacking detection algorithms with DIPPER (continued)
  + Here they are continuing to test AI detection algorithms when the actual AI content is paraphrased
  + They compare before and after a piece of text was paraphrased by DIPPER
  + Used Wikipedia prompts (I guess that means they took questions from Wikipedia and put them into AI generators)
  + The two variables they control in DIPPER are Lexical Diversity and Order Diversity, these specify how much the text should be changed from its original generation.
  + Generally, the trend is that as the diversity goes up, detection percentage decreases with all AI Detection algorithms
  + Semantic Similarity is how close two pieces of text are in terms of their meaning
  + This article also says that even though the semantic similarity of the text is very high, the AI detectors still cannot detect the paraphrased text at a high accuracy level
  + FPR: False Positive Rate
  + Generally speaking, the semantic similarity decreases slightly as the lexical diversity and order diversity increase. But still the percentage of semantic similarity of the paraphrased text is very high
  + 
    - Picture of the AI detection they did and how the AI detection faired when paired with paraphrased text
  + Semantic similarity: They found semantic similarity by using the P-SP model
    - This model was trained by a lot of different paraphrased text
  + P-SP score of 0.76 is what an average human paraphrased text would by
    - That is a good benchmark to see if text has its meaning preserved, if the P-SP score is higher than 0.76 it can be said that it is paraphrased (the semantics are preserved)
* 4.2 Models, datasets & detection algorithms
  + Continue here next time

9/26/2024

* <https://arxiv.org/pdf/2303.13408>
  + Background on detectors of AI-generated text (continued)
    - Watermark continued: Good watermarks are hard to remove and don’t affect the quality of the text
    - Statistical outlier detection methods: can also look at the statistical irregularities of the writing, meaning looking at the perplexity and entropy.
    - GLTR visualizer: uses statistics to tell humans if a text was made by AI
    - Closed-source GPTZero:
    - Open-source DetectGPT: AI generated text usually has a higher chance of being made by an LLM than an altercation/paraphrasing of the text
    - Classifier Methods: these models are trained to be able to tell the difference between human and AI written text. At the start it seems this was the approach to detecting fake news, Open AI made a model that uses classifier methods, its available online (uses human written text and text from 34 LLM’s)
  + Building a controllable discourse paraphraser
    - 
    - Problem with current detectors is when text is paraphrased
    - When text is paraphrased, AI detectors are tricked and the watermarks get removed and accuracy plummets.
    - What happens with a paraphrase is, the meaning of the text doesn't get changed (input semantics)
    - Making DIPPER: using a LLM “on a parallel dataset of paragraph-level paraphrases, which we modify to model control, external context and content reordering.”
    - Par3 dataset: This dataset has multiple different english translations of non english text into english, “aligned at a paragraph level”. These are thought of as paraphrases
    - Parallel data is two different translations of the same foreign language text
    - What they do is they align the different translations, at a sentence and paragraph level. They look at how the texts are similar and align the parts that are similar. Use semantic similarity scores from the paraphrase similarity metric in <https://aclanthology.org/P19-1427.pdf> making sure to use dynamic programming
    - Each sentence of one translation was aligned with a sentence or two from the other translation, aligned based on having similar meaning
    - Choose a subset of the alignments
    - Then they mixed up the sentences, making the order random. Calculate diversity control codes between the original paragraph and the shuffled second paragraph. “We compute lexical diversity (L) using unigram token overlap (F1 score), and the order diversity (O) using the Kendall-Tau correlation of tokens of overlapping words between pi ...pj and shuffle(qi ′ ...qj ′ )”. Then they take the values and normalize them to be in the set {0, 20, 40, 60, 80, 100}. 20 means there is a 20 percent lexical change from paragraph to paragraph.
    - 
    - Finally, they refine a T5-XXL LM that will give you p2p3 given q3q4q2 and the context of p1 and p4.
  + Experiments attacking detection algorithms with DIPPER
    - They are testing their DIPPER paraphrase against AI detectors and found that the text was able to escape detection across three LLMs (one being GPT3.5)
    - Used the AUC-ROC metric to judge how well the detector did
    - It's very important to have a near zero false positive rate, they set the false positive rate to 1 percent for their testing
    - They test on paraphrased test vs original AI generated text CONTINUE ON PAGE 5

9/24/2024

* <https://arxiv.org/pdf/2303.13408> (continued research on ways to detect paraphrased AI content)
  + Introduction:
    - AI text is a big problem as people are generating fakes that look very similar to actual human writing
    - Some of the more recent AI detection algorithms are watermarking, DetectGPT, GPTZero, and Open AI’s text classifier
    - The key issue here is, how do AI detectors respond when a smaller model paraphrases AI generated content, can they detect it?
    - It seems that the watermark AI detection watermarks and puts a green token on a certain word if it deems to be AI generated, then the detector looks at how many green watermark tokens there are in total to tell if the text is AI generated or not
    - The detector gets confused/doesn’t detect properly after Dipper (the articles paraphraser) paraphrases the content from ChatGPT
    - What does Dipper do?
      * They paraphrase by rewording longer text (like paragraphs), put the words/phrases in a different order, and leveraging the users input
      * Dipper keeps the text similar enough so that the meaning of the paragraph ins the same, while changing it so that AI detectors don’t know that its AI generated
      * Gives users the ability to control how much they want the text to be paraphrased (how different should it be from the original text in terms of lexical diversity (how many words in a text) and content reordering)
    - Misleads many AI detecting algorithms
    - How to account for this?
      * Retrieval methods (filter out only the important information from a large dataset)
      * They decide not to use watermarking or any statistics regarding the text
      * Use a LLM API provider
      * Stores every possible output value that the model made in a database
      * Semantic representation (ways words and their meanings is represented in the computer) of the text is then contrasted and compared with representations of every generation that is inside the database
      * Using the semantics (the meanings of the words), using both IR (information retrieval) methods (BM-25) and semantic vector representations like P-SP
        + <https://en.wikipedia.org/wiki/Okapi_BM25> (article on BM-25)
      * Retrieval based method/approach
      * Algorithm works well when given paraphrased text, detects 97.3 percent of text that was paraphrased from PG19 and 80.4 percent of text that was paraphrased from Wikipedia
      * 1 percent of the time you get false positives
  + Background on detectors of AI-generated text
    - Watermark: “a modification to the generated text that can be detected post-hoc by an algorithm while remaining imperceptible to human readers” CONTINUE RESEARCH ON THIS ARTICLE NEXT TIME

9/23/2024

* Looking at references in <https://ieeexplore.ieee.org/abstract/document/9260074> to get an idea of a starting point
  + Can’t find the GitHub anywhere in this article
* <https://arxiv.org/pdf/2303.13408> (article on retrieval as a way to detect AI generated text that has been modified from its original thing), there is a github repository on this link if you google “Dipper”



* + When people do something like this, it becomes hard for detectors to tell if it's AI written or not, so I want to address this.
  + DetectGPT’s AI accuracy goes from 70.3% to 4.6% when you use Dipper to paraphrase the language of the AI content. This happens without much real change to the actual content of the stuff that was AI generated
  + “To increase the robustness of AI-generated text detection to paraphrase attacks, we introduce a simple defense that relies on retrieving semantically-similar generations and must be maintained by a language model API provider”
  + “Given a candidate text, our algorithm searches a database of sequences previously generated by the API, looking for sequences that match the candidate text within a certain threshold. We empirically verify our defense using a database of 15M generations from a fine-tuned T5-XXL model and find that it can detect 80% to 97% of paraphrased generations across different settings while only classifying 1% of human-written sequences as AI-generated. We open-source our models, code and data”
* <https://www.proquest.com/central/docview/3101842024/363FBD02EBF34200PQ/3?sourcetype=Scholarly%20Journals&parentSessionId=tB7P9jMVnhkhu91QgSN32JT6APv39hikxPmXNsn0xjg%3D> (I am going to also read through this article, I read some of it yesterday, it's about how people can evade AI detectors by paraphrasing AI content), I might read this one later this week

9/19/2024

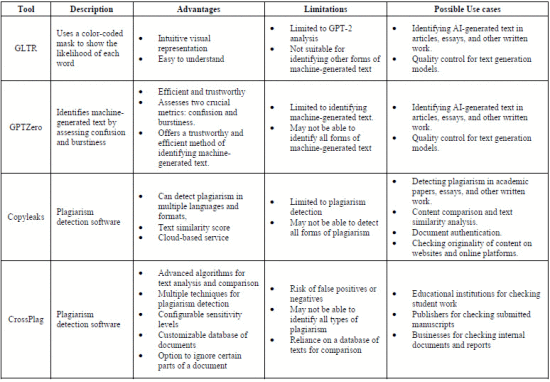
* <https://ieeexplore.ieee.org/abstract/document/9260074> (continued research into AI detector)
  + Shallow and deep learning models
  + They used many models including “SVM with radial basis function kernel, K-Nearest Neighbor, and Random Forest.”
  + L2-Regularized Logistic Regression had the most success among all the models
  + They also used a feed forward deep neural network
  + They used a maximum of 50 words for their tweet detection
  + They also used scikit-learn for their logistic regression, BERT model from Hugging Face
  + Also used pytorch for the deep neural network
  + “ Its training was carried out using mini-batch Adam optimizer with batch-size 32, the initial learning rate of 0.5, and the same exponential decay factor value”
  + They made the validation data set from randomly sampling 20% of the tweets from the training data. They used this for both the shallow and deep learning models
  + Logistic regression model: they chose the model with the highest weighted F1 score on the validation set. For each model, this took them about 60 minutes to do.
  + Deep learning models: they measured loss and stopped early if necessary
    - Used 10 epochs
    - Took about 5 hours for each deep learning model
  + Baseline features are without any other indicators and the other one is proposed features
    - They evaluated the model on both of those
    - They calculated a weighted F1 score
  + They found that the model with proposed features has a better score than the baseline, using statistics and the alpha = 0.01 significance level.
  + One of the reasons they found linear regression to be a better predictor is that the weights are brought down to zero as much. The reason for this is the L2 regularization.
  + Seems that linear regression is better to use here than deep learning network
* <https://www.proquest.com/central/docview/3101842024/363FBD02EBF34200PQ/3?accountid=34939&sourcetype=Scholarly%20Journals> (ways to bypass AI detectors)
  + Using text that is changed to avoid being detected, trying to see current AI detectors power
  + Many students like to do this, so it's important to see the accuracy of the AI detectors
  + Much use of AI today in the workforce and with school
  + People that don’t speak english as a first language seem to be more likely to be detected by AI detectors even if they didn’t use any AI tools
  + The goal should be to promote fairness in assessing a students work, not falsely accusing them of cheating
  + They want to look at disadvantaged kids writing as well as English Second Language students and see if they're writing is detected
  + Also looking at how much the AI detectors work when AI work is altered even a slight bit, that is expected to throw off the accuracy of the AI detector
  + The main question is are AI detectors reliable? Are they able to detect AI generated text enough to where they can actually be used in education settings by teachers?

9/17/2024

* <https://dl.acm.org/doi/10.1145/3512732.3533584> (Automatic Detection of Bot-generated Tweets)
  + This article looks at detecting bot generated tweets
  + LSTM, BERT, Char-RNN can be used for classifying fake tweets
  + The full article is behind a paywall, cannot seem to access it
* <https://ieeexplore.ieee.org/abstract/document/9260074>
  + BERT based AI detection bot on tweets that were written by AI
  + Seems that logistical regression does better than Deep neural network in this article
  + The question is about authenticity, so many tweets are out there and you want to figure out which of them are not fake
  + A lot of people share and believe fake news that these bots spread
  + Bots are autonomous and coded to spread fake news (bots have AI in them)
  + The goal here is to use an ML algorithm to distinguish between human and AI written tweets using the tweets content
  + BERT BASE and emoji (emoji2vec embeddings) detection was used. They also used URLS, hashtags, mentions
  + Related Work:
    - Random Forest seems to have the bet bot detection in a 2019 competition
    - <https://www.foi.se/download/18.46c3b221170ccb3336f1c6/1585826754727/Supervised-classification-of-twitter_FOI-S--6118--SE.pdf>
    - Deep learning models do better than shallow learning models
    - Typical models used include Convolutional Neural Networks, Recurrent Neural Networks, BERT, and other models
  + This specific study used a Logistic Regression and Deep Neural Network models
  + “The original data set had 100 tweets per user. Every user was labeled as either bot or human and additionally as male or female in the case of the human class.”
  + This study uses tweet based classification, each tweet was given a class (either human or bot)
  + In exploratory data analysis, they used an NLP cleaning pipeline made of “language detection, tokenization, word normalization, POS tagging, and lemmatization. ”
  + This study also performed the Latent Dirichlet Allocation (LDA). which “ is an unsupervised clustering algorithm for document topic identification. With the LDA method, we obtained word distributions for each topic. Afterwards, we estimated the posterior probabilities *P*(*Class*|*Word*) for each class with a MAP estimator:
  + P(Class=c|Word=w)=
  + C(c,w)+1
  + C(w)+|Word|”
  + It seems like they were able to classify by words and then associate certain words with bots and other words with humans
    - Words related to recruitment/jobs are most commonly generated by bots
  + Politics seems to have a large number of humans that tweet about it, yet there is a good amount of bots that tweet about it as well
  + How did they actually create the model?
    - First Approach
      * Used word2vec embeddings
      * Used token vectors and put them into the shallow ML model
      * Gave good results but not as good as the second approach they used
    - Second Approach
      * BERT model
      * Worked better than first approach
      * CLS and SEP tokens were applied
      * CLS is a classification token put in the start of the tweet
      * SEP is put in as a delimiter between sentences
      * The model schema includes first cleaning the tweet, then putting it into the BERT base, then contextual embedding, finally using a deep neural network with many layers and also a logistic regression to classify the tweet.

9/16/2024

* Still doing research into different AI detectors from <https://ieeexplore.ieee.org/abstract/document/10099219>
* Later, look at different links in the references of this article and also look at different stuff on Google Scholar), <https://dl.acm.org/doi/10.1145/3512732.3533584> (another good link on different ways to make AI detectors)
  + Copy Leaks continued
    - Checks the originality of the person's work
    - CopyLeaks can check the work from a variety of file formats: PDF, HTML, Microsoft word (add these different file formats to your product as well, probably going to make a website. Also can be integrated into learning management systems like Schoology
  + D. Crossplag
    - “ It utilizes a variety of techniques, including text content comparison, sentence structure analysis, and natural language processing algorithms, to detect instances of plagiarism.”
    - In text content comparison, D. Crossplag compares the content of the text to other online pieces of writing to see how similar they are
    - They are also able to use sentence structure analysis by looking at the similarities between the text and different websites
    - Natural Language Processing (NLP) uses NLP techniques to find similarities with various other documents
    - It checks one document against various other documents and finds similarities.
  + Comparing the different detectors
    - One problem here is that people can change their code/doc around and the detector might not be able to tell its AI generated anymore
    - Big need for better AI detectors as people are using AI more and more these days
    - Using computational intelligence methods is an option to improve the detectors accuracy
    - Comparison of different detectors from the website



9/12/2024

* Another Idea: College Admissions helper (or maybe just helping students determine good ideas for essay prompts)
  + <https://www.collegeessayguy.com/blog/college-essay-examples>
  + <https://blog.collegevine.com/ivy-league-essay-examples>
    - There are plenty of college admission essays available to run my model through
    - These are Common App essays and ivy league essays, I also see essays for different colleges available
    - I can use these essays and analyze them in order to tell students what type of essays they should write
    - Seems difficult to do because it's hard to find information about these college students, their info is private (extracurriculars, classes taken, etc)
* More about trying to create an AI detector
  + Use Google Scholar to look up research articles as a starting point
  + Use supervised learning (ML concept)
  + <https://ieeexplore.ieee.org/abstract/document/10099219>
    - AI’s use natural language processing in order to create human like responses
    - AI generators usually use supervised learning models like transformer-based language model and unsupervised text generation models
    - AI’s learn the patterns of language
    - AI’s can tend to give you text that is offensive, inaccurate, or biased. There tends to be information that comes from AI’s that are wrong.
    - AI’s can also give you writing that might be copyrighted somewhere online
    - There also can be information it gives you that is factually wrong
    - One way to create an AI detector is to train the model on a data set labeled as Artificially written or human written
      * Generative pre-trained transfer
      * Deep neural networks
      * Convolutional Neural Networks
      * Recurrent Neural Networks
    - Giant Language Model Test Room (GLTR): Using language models to detect AI content
      * This predicts what GPT 2 would say at a certain word and sees if the current text matches that
      * How GLTR works: “By examining the model's ranking of the words, GLTR applies a color-coded mask on the text that reflects each word's positional rating. All other words are highlighted in purple, while those that fall within the top 1,000 most probable predictions are highlighted in red, those that fall within the top 100 are highlighted in yellow, and the top 10 most likely predictions are highlighted in green. This visual representation allows for a direct and intuitive understanding of how likely each word was under the model's prediction.”
    - GPT Zero
      * Two metrics: confusion and burstiness
      * Looks at the amount of predictability in the text
      * Looks at the variance in the sentences, typically AI generated text has less variation and human generated text has more variety
    - Copy Leaks
      * They scan the text from all different documents including websites, documents, and databases to see if it is copied from somewhere
* <https://dl.acm.org/doi/10.1145/3512732.3533584>
  + Article on different ways to make AI detectors

9/10/2024

* <https://edintegrity.biomedcentral.com/articles/10.1007/s40979-023-00140-5>
  + This is a study on the different detection tools and this study tests each of their effectiveness in detecting AI generated content
    - Tested the effectiveness of OpenAI, Writer, Copyleaks, GPTZero, and CrossPlag
  + This article talks about how AI detectors right now tend to give false positives and are inconsistent
  + Seems like GPT 4 detectors are even less accurate than GPT 3 or 3.5 detectors
  + “Furthermore, the developers' tests reveal that the classifier accurately identifies 26% of AI-written text (true positives) as "likely AI-generated" while incorrectly labeling 9% of the human-written text (false positives) as AI-generated”
    - The percentages of AI detection aren’t great right now
    - This isn’t a great tool, its best to use as a supplementary check for AI and not use as an end all be all as it's not very accurate
  + “it is evident that the tools' performance was not entirely reliable. While some human-written content was correctly classified as "Very unlikely AI-Generated" or "Unlikely AI-Generated," there were false positives and uncertain classifications.”
    - We also see here that the AI detections seem to provide false positives (the detection software thinks its AI but its actually not AI)
  + This is a big issue as students' integrity is getting called into question even if they wrote it themselves.
  + “While the tools were generally more successful in identifying GPT 3.5-generated content, they struggled with GPT 4-generated content and exhibited inconsistencies when analyzing human-written control responses.”
    - Seems like the detectors are pretty good with GPT3 and 3.5 but struggle with GPT 4
  + There is still a level of uncertainty that requires further improvement in AI detectors. They aren’t reliable enough right now to use in really important situations, there is growth needed in this area
  + The way AI detectors work right now, they cannot be used as the single determinant for whether code/writing has AI. Humans have to manually check the code/writing
* I think the biggest problems might be the false positives, where students are falsely accused of cheating even when they didn't cheat at all.
* Also there's something to say for work that is written by a student but edited for clarity by someone else, that gets flagged for plagiarism
* Also the AI detectors we have today have a minimum word/character count, if you don’t reach that threshold the results aren’t reliable
* Telling the difference between AI and Human generated content
  + More word variance, smooth transitions from ideas, a lot of repetition and there isn’t much originality in AI generated content
  + It is kind of predictable in the way that it is general and on the surface when compared to humans writing
  + Does not have that same human touch
  + More basic, even if the writing style is better the AI generated text is very basic
  + There tend to be factual inaccuracies with AI generated content if your using a book or something
* <https://www.matellio.com/blog/ai-content-detection-tool-development/>
  + This outlines the steps for generating an AI detection tool

9/9/2024:

Ideas to research more into:

* Research into how you're going to create a detector to check whether code/writing was written by ChatGPT or other AI’s (what datasets are you going to use and how are you going to achieve greater accuracy than the normal AI detectors?)
  + <https://sites.google.com/fcpsschools.net/tjlibraryresources/home>
    - This is an AI code detecting website, although it doesn’t seem very well used or mainstream
    - Doesn’t seem like C++ is even supported for detection, I could try to do Java, Python, C++, JS other languages
    - Very, very slow in detecting 3 or 4 lines of code, needs improvement
    - It detects code if you straight paste it into the detector, but I was thinking that kids mostly disguise their code before they submit it, so can the AI detectors we have right now detect this? Also is there a case of false positives happening (code that is detected as AI generated but was really written by humans)?
* Research into college applications and how you can access data for giving students recommendations for how to make their application better
* Research into how to take a food, scan it in and identify

9/5/2024

* More into different ideas for senior research
  + More on concussions
    - <https://www.nih.gov/news-events/nih-research-matters/how-football-raises-risk-chronic-traumatic-encephalopathy>
      * Concussions can lead to CTE and eventually death, repeated head trauma puts players at huge risk for CTE
      * CTE risk can increase significantly with playing more years of football, as your at higher risk for suffering these injuries
    - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6760664/>
      * “Increasing impact velocity was associated with increased peak linear acceleration.”
      * Most of the more severe head injuries result from the higher impact velocity
  + AI that helps suggest what you should do in order to enhance your college applications
    - <https://www.sups.ai/>
      * This is an AI that helps students write essays and gives them suggestions in their college applications
      * It doesn’t write anything for the students, it just gives them tips
      * I was thinking of something similar to this but instead of only giving feedback on essays I was thinking about something that gives feedback on your overall portfolio
    - So like this, say you can link it with Common App and it tells you what to add to your portfolio in order to get admitted to certain colleges
    - It will give suggestions on what you need to do to give yourself the best chance at getting admitted to a certain college
      * Tells you about what extracurricular activities your need or tells you what SAT scores to aim for, it uses the information you have given it to tell you how to make the best college application
  + A better AI detector
    - I was thinking that a lot of AI detectors these days are very poor.
    - The variability in detecting whether ChatGPT Was actually used is very high
      * Some detectors will say 40% GPT, some say 50% GPT, some say 10% GPT, some say no GPT on the same essay. I have put the same words into these detectors and this is the result I've gotten.
      * Too much inconsistency right now
    - I would like to create something here that can better detect if a writing has come from AI, something that has more accuracy than the current detectors. I can even test it against the current detectors we have right now.
    - Maybe I can use the current AI detectors somehow to make an even better one hopefully, and I would want to create something that detects all AI’s not just ChatGPT, so I would include other types of AI in my detection
  + College email filter
    - Nowadays seniors get a lot of emails from colleges, some helpful and some not so much
    - This could filter through all the college emails you get and determine which one are helpful and you need versus ones that are unnecessary
    - For ex: I get around 20 emails a day for colleges and around 16 of them are emails I never read. They are from college I have no interest in applying to
    - Could save students a lot of time, makes them not have to read through unnecessary emails
  + Homework organizer
    - Something that you input your work into it and tell it due dates and everything your thinking of, and it will create a schedule for you and tell you what to do when
    - Reduces stress for students on planning out how to do their homework
    - Also I can have a person put in any events/extracirruclar activities they have on there and the AI will tell them how to manage their time best
    - TJ students could definitely find this useful
    - Maybe you could also tell it the sports games you want to watch and it will schedule your HW around that also
    - You can give it feedback on how well its doing in managing your time
  + AI that plays and wins Monopoly
    - Play against it and see what the win percentage is
    - See how good it is at strategizing against normal human players
  + AI that detects whether code was generated by AI or not
    - Similar to many of the writing tools we have but for coding
    - Tries to see what part of code was written by AI and what part was taken from the web
    - Very useful tool as AI is on the rise and students have been using AI a lot recently
    - It's hard to tell whether AI is used or not so we need to create a detector to tell

9/3/2024:

* More about senior research ideas (trying to answer Ms. Yilmaz’s comment questions):
  + <https://docs.google.com/document/d/1fMeQLRpmpyVNlL3W0ntQ_zOHMq9mXcq1a3t3Xiq3QFo/edit>
    - This talks about estimating speed using Computer Vision
    - They used YOLO and tracked the speed of cars going down a flat road
    - Can take this along with the force required to make a concussion and use that
  + <https://www.henryford.com/blog/2022/02/head-injuries-when-to-see-a-doctor#:~:text=%E2%80%9CThe%20most%20dangerous%20place%20to,head%2C%20just%20above%20your%20ears>.
    - Side of the head is most dangerous location for a concussion above the ears
  + [https://www.flintrehab.com/areas-of-the-brain-most-vulnerable-during-a-traumatic-brain-injury/#:~:text=Of%20the%20four%20main%20sections,frequently%20by%20traumatic%20brain%20injury.](https://www.flintrehab.com/areas-of-the-brain-most-vulnerable-during-a-traumatic-brain-injury/#:~:text=Of%20the%20four%20main%20sections,frequently%20by%20traumatic%20brain%20injury)
    - cerebral cortex is affected by brain injuries (TBI)
    - Out of the main parts of the cerebral cortex, the Frontal lobe is hurt most often by TBI’s or concussions. Most of the time you hit the front side of the head, and that can cause concussions.
    - If the concussion affects the primary motor cortex which is inside the frontal lobe
    - Other parts of the brain and body can get affected by the concussion
    - If you get a concussion and it affects the hippocampus, you can have difficulty remembering things
    - If it affects the basal ganglia you can start to move irregularly
  + <https://www.urmc.rochester.edu/news/story/single-brain-region-is-key-to-assessing-the-impact-of-repetitive-head-hits-concussions#:~:text=Prior%20research%20involving%20football%20players,injury%2C%20as%20well%20as%20CTE>.
    - The midbrain is also a part of the brain to watch when one has a TBI
    - No matter what specific part of the brain a concussion hits, the impact of the force goes into the midbrain.
    - Imagining into the midbrain can help with diagnosing a concussion
* More research into general ideas
  + <https://www.aiacceleratorinstitute.com/7-problems-artificial-intelligence-can-help-you-solve-in-your-company/>
    - Ideas that AI can help solve
      * Text and NLP analysis: look into the tone of a text and whether it is positive or negative
      * Intent Prediction: Use customers previous clicks and data to determine what they are going to do next
      * Data analysis: Take data and analyze it to make some meaningful predictions
      * Fraud: Detect fraud or fake stuff on the internet or otherwise, can use ai for this and determine the accuracy of the model
      * Marketing
      * Fantasy football and sports betting helper: I need to research more into this and see if there is anything on the market that bets well already

9/2/2024:

More ideas

* Allergen listing website for all restaurants
* Google Maps enhancement where you see the lane you're supposed to be in at all times

8/29/2024:

More research into previous ideas:

Concussion detector:

* + Location of the head and concussion risk <https://www.stevens.edu/news/researchers-show-how-side-hit-head-could-damage-brain-lead-concussion>
    - Corpus callosum area is very at risk for concussions
    - When hit hard on the side of the head, the head will rotationally accelerate this causes vibrations in two parts of the brain, the corpus callosum and the periventricular area
    - These two regions seem to be most important when it comes to getting a concussion, these two regions seem to have a lot of cell and tissue damage when concussions happen
  + Seems that it can occur in helmet to helmet collisions or when the head hits the ground hard
  + Seems that you can get a concussion only from getting hit 5-15 mph if its a low speed car crash
    - <https://bermanlawyers.com/concussion-without-hitting-your-head/#:~:text=It's%20also%20important%20to%20remember,serious%20damage%20to%20the%20vehicle>.
  + The way a concussion gets caused is when you have an injury to the skull
    - <https://www.pennmedicine.org/for-patients-and-visitors/patient-information/conditions-treated-a-to-z/concussion#:~:text=A%20concussion%20can%20result%20from,lose%20alertness%20(become%20unconscious)>.
    - Sometimes the concussion leads to that person being knocked unconscious (around 5-10%) but many times that doesn’t happen
    - “In a severe impact to the head, the brain moves and hits the skull causing injury. During a boxing match, the brain moves from side to side after the impact of a punch. Following a concussion head injury, confusion and disorientation due to temporary distortion of the brain may result.”
      * <https://www.pennmedicine.org/for-patients-and-visitors/patient-information/conditions-treated-a-to-z/concussion#:~:text=A%20concussion%20can%20result%20from,lose%20alertness%20(become%20unconscious)>.
* <https://www.mdpi.com/1424-8220/22/15/5592> (continue to look into this link)

Homeless people app:

* Would need to be able to tell if someone is actually homeless or not
  + Signs:
    - Living on the streets
    - Clothes that are ripped up
    - People that don’t have a place that they call home
    - <https://ed.sc.gov/policy/federal-education-programs/essa-title-ix-part-a-mckinney-vento-homeless-assistance-act/potential-warning-signs-of-homelessness/>
      * “Poor hygiene”
      * “Poor health/nutrition”
      * “Lack of continuity in Education”

Model that helps decide if people like movies or not:

* Netflix (https://help.netflix.com/en/node/100639):
  + Uses your information on what you browsed to come up with new movie recommendations
  + Uses what other people liked that are similar to you
  + Uses characteristics of the movies you watch to give recommendations
  + They also input when you watch Netflix (time), the languages you watch, and other stuff

8/27/2024:

More ideas for senior research project:

Sports Ideas:

* AI that helps basketball general managers decide on players. Nowadays spending efficiently in the NBA is paramount and I think making a project to help gm’s will be really important and new. Each team is faced with unique circumstances regarding how much money they can spend and how many draft picks they can use. I can make something that helps those GM’s make decisions about those players.
* **Concussion detector AI**: in the NFL head injuries are a huge concern. I could try to make something that helps teams identify when a player has a concussion. This would be really helpful as teams need to take players out when they have a concussion because staying out longer could pose health risks. Additionally, it's difficult for humans to tell immediately if a player has a concussion so I could try to train my model to detect those concussions.
  + <https://www.webmd.com/brain/ss/slideshow-concussions-brain-injuries>
    - Hard hit to the head causes the concussion
  + <https://www.google.com/search?q=how+to+tell+if+a+football+player+has+a+concussion&rlz=1C1WDIF_enUS1073US1073&oq=how+to+tell+if+a+football+player+has+a+concussion&gs_lcrp=EgZjaHJvbWUyCQgAEEUYORigATIHCAEQIRigATIHCAIQIRigATIHCAMQIRigATIHCAQQIRifBTIHCAUQIRifBTIHCAYQIRifBTIHCAcQIRifBTIHCAgQIRifBTIHCAkQIRifBdIBCTEzNjIxajBqN6gCALACAA&sourceid=chrome&ie=UTF-8&safe=active&ssui=on>
    - Appear dazed or stunned or have difficulty focusing
    - Get confused about what they are supposed to be doing on the field/court
    - Forget plays
    - Look unsure of game, score or opponent
    - Move clumsily (appear off-balance or dizzy)
    - Be slow to answer questions
    - Lose consciousness (only occurs in about 10 to 15 percent of concussions)
    - Experience behavior or personality changes
    - Be unable to remember events prior to or after the injury
  + <https://www.brain-injury-law-center.com/blog/concussion-symptoms/#:~:text=Specifically%2C%20concussions%20occur%20at%20approximately,more%20than%20normal%20gravitational%20force>.
    - concussions occur at approximately 90-to-100 g-force, meaning 95 times more than normal gravitational force.
* Virtual Reality Training environment: Creates a simulation of the game for NFL players. They can use this for their training in the offseason, it will help them as they are training on actual game situations just through a virtual reality machine. Instead of going to preseason games, NFL players can just use this to train when they aren’t playing. I was thinking it puts the player right in real game situations where they feel like they are playing and living an actual NFL play.

Random Ideas

* **A model t**hat can use movies a user already watched to determine whether they will like a movie. The model gives the user a confidence score of how confident they are that the user will not like the movie. The user uses this in order to not waste time watching movies they don't like. I know this could be helpful because I tend to get mad when I watch a movie that I don’t like, it seems like a waste of time for me.
* Better Customer support program. These days it's very annoying talking to customer support and it's difficult to get your questions answered. With this model I can try to help those customers. Many people are annoyed with customer service and they think it is a waste of time. With this model people can get the questions answered more quickly and efficiently without wasting as much time.
* A program that helps senior citizens do their taxes and helps them read bank statements and other things that come in the mail. I was thinking that for older people that have a hard time seeing or anything else doing their taxes and reading all this mail will be difficult. With this program I can help those people do their taxes and read bank statements more efficiently.
* AI that helps locate people that are homeless/without food and gives them food. I could make an app that finds the location of people that need food and that tells organizations where to give their food out to. I feel like this would be really helpful because most people don’t know where homeless people reside. I would need to get data and user input in order to do this.
* Detecting unnecessary emails/advertisements. I know that there already is a spam folder in email accounts but I still feel like there are a lot of spam emails that reach my inbox. I was thinking of making something that can be installed in someone's inbox that will detect spam/advertisement emails. It should detect any unwanted emails.

8/22/2024

Thinking about more ideas for Senior Research Project:

More Sports Ideas:

* Idea 1: A model that studies footage of basketball players and teams and tells you what plays to run. It could be something that coaches use in key situations in the game. Right now coaches will just use their instinct to make play calling decisions, but with this model they will have a more calculated and accurate play to run. I was thinking NBA coaches could use this model and I could train it on data of NBA players. After a lot of training hopefully it will give coaches better plays to run than they would have thought of themselves.
* Idea 2: An automated strike zone for baseball. Right now human umpires call balls and strikes in baseball and it has led to much controversy because of human error. A lot of times these umpires make the wrong call but it's really difficult to judge them because the ball comes so fast at them. Instead of these human umpires, I can try to make a robot umpire that identifies whether it's a ball or strike and relays that information to a person who then makes the call. The advantage to this is that human error is taken out of baseball and all the calls will be correct.
* Idea 3: Strategical bot that tells goalies where to go in penalty shots in soccer. It will analyze data and tell the goalie where they should jump to. I'm thinking it would be like an earpiece that the goalie wears. Based on the players history and the way they are running up and taking the penalty, the bot will tell the goalie to jump either to the left or right or stay in the middle. The bot can also tell the goalie whether to jump up and guard the top part of the goal or stay down and guard the bottom.

Random Ideas

* Idea 1: Some kind of app that lets you try on clothes on yourself. I think it's really helpful to see how something fits on you so you don’t have to worry about going to the store. I'm not a big fan of going to the store anyways so I would definitely like this. I don’t think there's anything like this in the market right now. My thought is that you could just open the app and take a picture of the clothes then you enter in your measurements and it will show you how it fits.
* Idea 2: Mind controlled board game player. Takes what the mind says and executes that in a game. I'm thinking about chess or something like that. It could also think for itself and advise the user to not do certain moves that could cause a loss. I saw how people tried to do a mind controlled car so I was thinking this would be something similar. I also think this is something that is really innovative because it could help older people that might not be able to move their hands well play board games.
* Idea 3: Movie dialogue predictor, takes in dialogues from movies and predicts the movie that dialogue was from. I think this is an interesting idea because people sometimes wonder “what movie was that dialogue from?” and with this product I could solve that problem. It would need a lot of training on various movie dialogues and eventually it would be able to identify movie dialogues well. I could package this into an app.
* Idea 4: An app that will allow you to take pictures of a product and it will tell you the lowest price for that product across the market. So for example if you go to a store and take a picture of Jordans, the app will tell you the place where you can get the best priced Jordans and it will tell you directions to get there. You will probably also have to add a filter and tell the app within how far you want to search for the product.
* Idea 5: Design a website/extension that is similar to grammarly but for coding. On this website you will be able to code and as you code it will give you suggestions on how to improve your code and make it more efficient. For example it will give you a suggestion like “make these 4 if statements into 1 for loop” to help with readability. I think this website could help with readability, efficiency and writing better code. It will help students write code but not tell them how to do it so they can use it in school also.

Ideas that are similar to what we learned in AI (train with some input data and some output data, then the AI can take in input data and predict what the output data will look like using a NN):

* Idea 1: Student handwriting recognition. You can use this to make sure that a student's work is their own work and not anyone else's. An English teacher can use this to grade essays, let's say they put a student's work in the recognition software and it can tell whether that's the same student's handwriting or not. It will help in detecting whether a student turned in another person's work.
* Idea 2: Gift suggestion app. You enter in a person's characteristics and this app will tell you what to get them. This app will give you a link to those websites as well. I think giftgetting is definitely a difficult task and this app can help those that are struggling with getting gifts for their loved ones. From experience I know that this will be helpful to people that aren’t great at shopping or don’t know how to shop. Plus it will save them time as they don’t have to spend hours at the mall.

8/20/2024

Ideas for Senior Research Project:

Idea 1: Design a better map. Right now Google Maps gives you a 2D view of the street but I was thinking of a 3D version of that. I find it difficult to navigate using Google Maps so I think a 3D navigation app could definitely be helpful for people that struggle with navigating. I'm thinking of using it for walking around in cities or in school. In the app, you would have 3D directions showing you exactly where to turn and go.

I can also create a 3D map for TJ. I know a lot of people struggle with navigating the school in 9th grade and a 3D map that shows directions to your classroom would be helpful. I'm thinking that once you open the app, you should be able to put a classroom in TJ in and it will show you camera footage or a really good model and show you exactly where to turn left/right in the camera so you have no confusion.

Idea 2: Design some kind of algorithm that determines the optimal amount of rest for NBA players. Nowadays, NBA players are facing the question of how much they should rest in the regular season in order to be ready for the playoffs. It's really important to take the regular season seriously while also making sure that you are fresh for the playoffs. I want to make some model that takes into account the players age, injury history, team, teammates and tells us how much that player should rest in order for the optimal season. Although I am not sure how I can measure that.

More ideas:

Image detection/Predictions (games?):

* Idea: AI that plays Jenga. (I would need an engineering team assistant for this). Design a robot that can play Jenga and knows where to place the blocks to try and win. I would try to do the software part of this but I'm not sure if it's practical because it's going to be hard to figure out which block to take out at a given time.
* Idea: AI that plays table tennis. (I would need an engineering team assistant for this). I saw this project a couple years ago, I don’t think it was fully developed. I think it would be very interesting to see a computer with a robotic arm that plays table tennis. I would be curious to see if it could beat humans.

Sports Predictions:

* Idea: Make a product that can help people win while betting on football. Nowadays many people bet on football but not many people know how to bet the right props. On average, people end up losing money. With my product I would try to find the best bets and suggest it to the customer. I could then compare my results with my model to an average betting person and see which one is better.
* Idea: Make a product that manages a fantasy football team. Nowadays many people play fantasy football but fantasy football is also very unpredictable and people don't manage their team well. With this model I can try and fully manage a fantasy football team for a person. I'm curious if this model can play fantasy football better than I can or an average person can.

Convenience:

* Idea: Translator app. Something that can detect speech and translate it to the user's desired language (starting with English only). It can help those that don’t know how to speak English communicate with English speakers.

Food:

* Idea: Make something that can detect the ingredients in a food based on a picture/video/scans of the food. I think that for me, as someone with allergies it's really important to know what ingredients are in food. I know that it would be difficult to create but I feel like it would be worthwhile and really helpful for anyone with allergies. I would have to take some kind of a scan of the food and then my model would have to predict the ingredients in the food.