|  |  |  |  |
| --- | --- | --- | --- |
| **Project Title** | Plagiarism detection | | |
| **Project Code** | AIS411 | **Course Name** | Natural Language Processing |
| **Professor** | Ensaf Hussein | | |
| **TA** | Ziad Elshaer | **Mentor Name** | Ziad Elshaer |
| **Team Name** | Team alpha | | |
| **Team Members** | Ahmed Kamal | Youssef Hawary | Mohamed Abdelnaser |
| Omar Morshdy | Text. | Text. |
| **Problem Summary** | Rapid programming improvements have made it more difficult to identify code plagiarism and AI-generated material, especially because of complex refactoring and transformation techniques that circumvent conventional detection systems. The majority of existing techniques are unable to distinguish between code that is generated by AI and that is written by humans, or to detect complicated instances of plagiarism. These difficulties make professional responsibility more difficult and compromise academic integrity. In order to solve these problems, the paper suggests a novel framework that combines many cutting-edge models with the goal of greatly improving the precision and efficacy of code authenticity evaluations in both the professional and educational spheres. Using advanced machine learning techniques, this method not only detects different levels of plagiarism but also successfully separates codes created by AI from those created by humans. | | |
| **Methodology** | The methodology adopted here is a dual-framework approach that utilizes state-of-the-art machine learning models in tackling the highly complex issue of code plagiarism and AI-generated code detection. Within the Plagiarism Detection Framework, three transformer-based models are integrated: UniXcoder, GraphCodeBERT, and CodeT5. These models, carefully fine-tuned to a great extent, are capable of finding different levels of transformation in code, from replicas to heavily modified variants. An ensemble method aggregates the outcomes to determine the final classification by taking the highest probability values into account, which increases the detection accuracy over different levels of plagiarism.  In the detection of AI code, this approach leverages CodeBERT fine-tuned with few-shot learning and prompt engineering to improve the model's ability in discriminating between human-written and AI-written code. The process essentially trains the model on carefully selected examples to enhance its predictive accuracy without requiring extensive data. Further, the integration of LangChain for dynamic prompt engineering calibrates the model's sensitivity to subtle cues that separate AI from human code inputs.The deployment is supported by a Streamlit web app, offering an interactive interface for analysis and classification in real time. Such a deployment not only demonstrates the practical application of the models but also assures that users will be able to use the system with ease in many educational and professional environments. | | |
| **Achievements and Skills Gained** | 1.created and optimized a collection of pretrained models to detect code plagiarism and artificial intelligence-generated content with high accuracy.  2.developed expertise in integrating and using transformer-based models in a real-world, useful application, such as UniXcoder, GraphCodeBERT, and CodeT5.  3.learned how to use Streamlit to implement AI models, improving real-time data processing and user engagement.  4.Improved proficiency with quick engineering and few-shot learning to maximize AI performance with little training data.  5.Advanced understanding and real-world implementation of machine learning to distinguish between code produced by humans and AI | | |

|  |  |
| --- | --- |
| **Project Title** | Text. |
| **Main Results** | The ensemble model integrates outputs from UniXcoder, GraphCodeBERT, and CodeT5, providing class predictions across seven plagiarism levels (Non-Plagiarized, L1-L6).  The models' probabilities are summed and normalized using a softmax function to determine the most likely class. In a sample case, the highest probability indicated Level 5 (Logic Change) plagiarism, highlighting the framework's efficacy with an 88% accuracy rate, particularly for complex code transformations |
| **Discussion and Conclusion** | The ensemble model achieved an impressive 88% accuracy in detecting code plagiarism and 83% in identifying AI-generated code, significantly outperforming existing tools. The results highlight the framework's capability to address complex detection challenges, advancing the field of academic and professional integrity. |
| **References** | [1] J. Smith et al., ”Machine Learning for Source Code Similarity,” Journal of Computer Science, vol. 25, no. 3, 2022.  [2] L. Brown and K. Green, ”Graph-Based Representations for Code Plagiarism Detection,” ACM Computing Surveys, vol. 54, no. 2, 2023.  [3] R. Lee et al., ”Code Stylometry for Plagiarism Detection,” IEEE Transactions on Software Engineering, 2023. 16  [4] A. Kumar et al., ”Defeating Plagiarism Detection Systems with Automated Transformations,” arXiv preprint arXiv:2010.01700, 2020. [5] T. Harris and E. Davis, ”AI Code Detection Challenges,” Proceedings of the AI Detection Symposium, 2023. |
| **Future Work and Suggestions** | Future enhancements should focus on expanding the dataset diversity, incorporating more programming languages, and refining AI detection capabilities with advanced model architectures. Further development of the deployment interface to support multilingual capabilities and integrate more sophisticated visualization tools is also recommended. Work on more than 30 files of code |
| **Group Photo** |  |