INTRODUCTION
Blue Wave AI Labs builds actionable and ethical AI solutions that evaluate and mitigate risk, increase efficiency, and optimize performance for the nuclear energy and defense industries by transforming data into solutions. Blue Wave spends a considerable deal of time writing proposals to companies to lay out the terms of potential business together, which has incurred a fair amount of labor costs and lead time. Therefore, Blue Wave wants to automate the writing of proposals for their nuclear utility clients to reduce labor hours and improve lead time.

OBJECTIVE
Using open-source LangChain components, this project aims to develop a nuclear proposal chatbot that runs on private servers and does not share sensitive information with other entities on the Internet. Based on prompts and text-based interactions, the tool will generate and enhance proposals related to the nuclear energy domain using Natural Language Processing (NLP) for automation.

INTERACTION
One example of an interaction with the chatbot could be:
- User asks a question
- Chatbot provides an answer, or opts out of answering if the question is outside its domain of knowledge
- Question and answer are added to conversation history
- User is prompted to rate the chatbot's answer
- Thumbs up, thumbs down, no rating
- User is prompted for reasoning/comment if they gave a thumbs down
- Rating is saved to database

RAG Search
Retrieval augmented generation (RAG):
A method to retrieve information from large databases; used with a large language model (LLM). The trained LLM can generate a response to questions based on the results of the RAG search.

RAG (input question) --> RAG output
LLM (RAG output) --> final output

Works
- Used the 'all-MiniLM-L6-v2' and 'Zephyr' models to create chatbot.
- Implemented output scoring using reinforcement learning with human feedback (RLHF).
- Accessed corpus for database used in RAG search.
- Used Python libraries such as HuggingFace, Transformers, and LangChain.
- To create our final model, we tested many different HuggingFace models on smaller scales, and then chose the most efficient one to build on top of, and train on the corpus.

How They Work Together
- Web scrape to continuously update the corpus
- Process the scraped data to be usable for the model
- Train the chatbot on the corpus
- Ask a question to the chatbot
- RAG search looks through the corpus to find a range of possible answers
- Chatbot takes the RAG output and figures out the best answer for the user

Web Scraping
Web Scraping
The process of extracting content from a website.

Web Crawling
The process of discovering URLs on the website.

Government Regulatory Commission (NRC)
Government agency that regulates use of nuclear material.

Works
- Scraped NRC website for all documents
- Stored all documents
- Scraped documents and extracted text
- Stored all data to add to corpus

Methods
- Used Python libraries such as Requests, PyPDF, BeautifulSoup
- Used the NRC API to access the documents

RESULTS
- Fine-tuned the model to take less than 30 seconds to answer questions
- Expanded the corpus to further its nuclear domain knowledge, increasing accuracy of answers
- Chatbot saves conversation history, so it can better expand on previous answers

Conclusions
Implementing Web Scraping and RAG Search leads to a cohesive final product which includes a fine-tuned machine learning model that is capable of answering nuclear-related questions for Blue Wave clients. The model allows many people to be serviced at one time, and lessens the load of cost, time, and resources.

Future Work
RAG Search: 1) Optimizing chatbot to provide higher quality answers by further training of the LLM through RLHF. 2) Extracting data more efficiently by optimizing the algorithm. 3) Working on building a user-friendly GUI.

Web Scraping: 1) Fine-tune the model by constantly updating the web scraping function. 2) Web scraping more websites to give the model more data.

Acknowledgments
Our sincerest gratitude to Maria Berardi, Rizki Oktavian, Phil McCalley and everyone at Blue Wave AI Labs for their guidance and support in this project. We would also like to extend our utmost thanks to all the Data Mine staff that helped to make this project happen.

CONTRIBUTORS: