Objective

As an ML Engineer at Sleek, you'll always prioritise staying up to date with cutting-edge technology and working with great OSS models!

Your mission? Run a benchmarking analysis on two named-entity recognition (NER) models, SecureBert-NER and CyNER, using the DNRTI dataset. Once you're convinced of who's the best fit for our product, your next move is to ensure we can easily use it on-prem. This calls for crafting a Docker container to host the chosen model and all essential components.

To truly elevate user experience, you'll round it off by creating a user-friendly web interface using Streamlit. This interface will empower our customers to effortlessly upload text files and receive reports detailing all the discovered entities.

Tasks

Benchmarking Analysis

- ☐ Obtain the DNRTI dataset, a collection of documents with annotated named entities.
- ☐ Implement scripts to preprocess the dataset and prepare it for evaluation.
- ☐ Evaluate the SecureBert-NER [1] model on the DNRTI dataset [2]. Measure performance based on latency, precision, and recall metrics.
  Notes:
  - Depending on your computational resources, this might take significant time. Feel free to evaluate on a subset of DNTRI, but make sure to specify how you created the subset and why.
  - Reading the referenced paper (Ref. 1) isn't necessary for the exercise's success, but it could improve your understanding of the used model.
  - Ref. 2 can help guide you on how to perform NER evaluation.
- ☐ As the model is trained to identify distinct sets of entity classes, and the DNRTI dataset encompasses a separate set, please refer to the following class mapping when conducting your comparison:

DNRTI	SecureBERT-NER
HackOrg	APT
SecTeam	SECTEAM
Idus, Org	IDTY

OffAct, Way	ACT, OS, TOOL
Exp	VULID, VULNAME
Tool	MAL
SamFile	FILE
-	DOM, ENCR, IP, URL, MD5, PROT, EMAIL, SHA1, SHA2
Time	TIME
Area	LOC
Purp, Features	-

Create a NER (named entity recognition) service

- ☐ Create an architecture that allows users to send text to an API and receive the extracted entities and their class.
- ☐ Build a Docker container, as you see fit, to run this HTTP-based service. Ensure that you allow offline usage of your work (meaning your work should run entirely on-prem).
- ☐ Verify the functionality of the Docker container locally.

Bonus: Web UI Development using Streamlit / React

- ☐ Develop a web interface using the Streamlit framework, that includes a file upload feature for text files.
- ☐ Implement backend functionality to process uploaded files using the NER model. Only one file should be processed at a time.
- ☐ Generate a results table in the web UI containing two columns: class names and their identified entities.

Deliverables

- ☐ A benchmarking report and code for evaluation of SecureBert-NER model based on recall and precision metrics.
- ☐ Docker container containing the NER model and required dependencies.
- ☐ *Optional:* Streamlit web application providing a user interface for NER model usage, including file upload and functionality.
- ☐ Documentation detailing the setup, usage instructions, and any relevant considerations for each deliverable.

Submission Guidelines

Please ensure your completed assignment is submitted as a GitHub repository, including clear documentation and instructions. You should dedicate up to one day of work (8 hrs) to the exercise, spread out over a maximum of three days.

Note

A partial solution, showing off your way of thinking and capabilities, is better than no solution at all. Feel free to reach out for any clarification or assistance during the assignment.

Good luck!