

DP-Fusion: Token-Level Differentially Private Inference for Large Language Models

This repository contains the codebase for running **DP-Fusion**, a Differentially Private Inference (DPI) framework for document privatization, along with several baselines and attack methods.

The project contains:

- A set of **100 paraphrased documents**
- The proposed **attack and defense methods**
- Code for **mounting privacy attacks** using perplexity and min-k strategies
- **DP-Fusion** defense implementation

Directory Structure

```
.
├── all_methods_results.json      # Results from various methods
├── input.json                   # Parsed TAB-ECHR dataset with
entities marked
├── candidate_set_100.json       # Candidate entity list per
privacy group
├── Attack.py                   # Script for running attacks
├── DP-FUSION_Defense.py        # Script for running DP-Fusion
defense
├── environment.yml              # Conda environment
specification
├── outputs_qwen_public.json     # Output from public baseline
├── output/                     # Directory for outputs
└── README.pdf                  # You are here :D
```

output/ Directory

This directory contains the output files generated from running both the attack and defense pipelines.

Attack Outputs

- **test_attack_meta_data.pkl** Contains metadata from the attack, including the log probabilities assigned to tokens for each candidate.

- **test_attack.json** JSON file containing the results of the attack, including candidate paraphrases and their respective scores from different attack strategies (e.g., min-K, perplexity-based).
- **test_attack.pkl** A pickle version of test_attack.json, provided for more efficient loading and analysis.
- **test_attack.log** Log file capturing the stdout/stderr from the attack script execution.

Defense Outputs

- **test.json** Resultant file from running the DP-Fusion defense. It contains the redacted_text field with paraphrased outputs generated via the DP mechanism.
- **test.log** Log file for the defense execution, capturing console output and debug information.

Installation

To set up the environment, use:

```
conda env create -f environment.yml
conda activate dp-fusion
```

Running the Attack

Use the following command to run the attack pipeline (perplexity and min-k based):

```
python Attack.py \
  --input_file input.json \
  --start 0 \
  --end 100 \
  --gpu "0,1,2,3" \
  --model_id Qwen/Qwen2.5-7B-Instruct \
  --output_file outputs_qwen_public.json \
  --attack_output_file_json output/test_attack.json \
  --attack_output_file_pickle output/test_attack.pkl \
  --attack_output_file_meta_data output/test_attack_meta_data.pkl \
  --to_debug True \
  --number_of_cands 5 \
  --hf_token "hf_xxx" > output/test_attack.log
```

Attack Script Arguments

Argument	Description
--input_file	Input JSON file with parsed dataset
--start / --end	Document index range to process
--gpu	Comma-separated list of GPU indices
--model_id	HuggingFace model ID
--output_file	The file on which you want to mount the attack
--attack_output_file_json	JSON output for attack results
--attack_output_file_pickle	Pickle file for storing attack results
-- attack_output_file_meta_data	Metadata output file
--to_debug	Enable debugging mode (verbose)
--number_of_cands	Number of candidate generations per prompt
--hf_token	HuggingFace token (use a private token, not shared)

Running the Defense

Use the command below to run the DP-Fusion defense:

```
python DP-FUSION_Defense.py \  
  --input_file input.json \  
  --start 0 \  
  --end 1000000
```

```

--end 100 \
--gpu "0,1,2,3" \
--model_id Qwen/Qwen2.5-7B-Instruct \
--alpha 2.0 \
--output_file output/test.json \
--to_cache True \
--hf_token "hf_xxx" \
--beta_dict '{"PERSON":0.1,"CODE":0.1,"LOC":0.1,"ORG":
0.1,"DEM":0.1,"DATETIME":0.1,"QUANTITY":0.1,"MISC":0.1}'
> output/test.log

```

Defense Script Arguments

Argument	Description
--input_file	Input file with parsed dataset
--start / --end	Document index range
--gpu	GPUs to use
--model_id	Model identifier from HuggingFace
--output_file	Output file for generated text
--output_dir	Output directory (default: output/)
--experiment_id	Identifier for the current run
--alpha	Global DP noise scale
--beta_dict	Per-entity DP noise levels
--to_cache	Cache LLM outputs (uses more GPU memory)
--to_debug	Print debug info

Argument	Description
<code>--independent</code>	Use independent group-level privacy
<code>--max_tokens</code>	Max tokens to generate
<code>--hf_token</code>	HuggingFace token (keep private)

Notes

- `candidate_set_100.json`: Union of private entities across documents grouped by privacy types.
 - `outputs_qwen_public.json`: Output from the baseline public generation model, used for attack evaluation.
 - All attack outputs include `.json`, `.pkl`, and metadata for detailed analysis.
-