

MQuAKE-Remastered

MQuAKE-Remastered is an audited version of the [original MQuAKE dataset](#), designed to address up to 33% or 76% of corrupted questions and ground truth labels found in the original dataset. This project consists of four datasets: CF, CF-3K, CF-6334, and T. While CF-6334 is a newly created dataset (subsamped from CF), the other three are fixes applied directly to the original MQuAKE dataset. We provided a comprehensive and accurate benchmark for evaluating multi-hop knowledge editing methods.

Dataset Overview

- `MQuAKE-Remastered-CF-3k.json` : Our audited MQuAKE-CF-3k that is free of intra/inner contamination, conflicting edits, missing instruction information, and duplicate cases with a total 3,000 cases when applied with proper data utilities. This dataset is created to audit the issues in the original dataset and have the same use case as the original MQuAKE-CF-3k dataset.
- `MQuAKE-Remastered-CF-9k.json` : Our audited MQuAKE-CF dataset with a total of 9,171 cases.
- `MQuAKE-Remastered-CF-6334.json` : Our audited dataset (a subset of MQuAKE-CF) aiming to meet the need of parameter-based knowledge editing methods where learning is required. It has a total of 6,334 edited cases.
- `MQuAKE-Remastered-T.json` : Our audited MQuAKE-T dataset with a total of 1,864 cases.

Datasets with postfix of `T`, `CF-3k`, and `CF-9k` could be evaluated by memory-based editing method when proper masking is applied. But since the masking may not work with parameter-based methods, we created `MQuAKE-Remastered-CF-6334.json` that don't require any masking and are friendly to both memory and gradient editing methods. We provide a brief metadata at `metadata.json` and refer interested readers to our paper for more details.

Dataset Snapshot

Before everything, we refer our reader to the [original MQuAKE dataset](#) to form a better

understanding on its original design and intended tasks.

Keywords definitions

- `case_id` : The unique id of a single multi-hop question case.
- `requested_rewrite` : A list of edits (s, r, o->o*).
- `questions` : Three multihop question in natural language format.
- `answer` and `answer_alias` : The pre-edit correct label and alias of label.
- `new_answer` and `new_answer_alias` : The post-edit correct label and alias of label.
- `single_hops` : A list single hop question of the multi-hop question. Answers are for pre-edited scenario.
- `new_single_hops` : A list single hop question of the multi-hop question. Answers are for post-edited scenario.
- `orig` : A dictionary that provides more information regarding each hop (s, r, o/o*) and the id of relevant entities and relations.

Example:

```
{
  "case_id": 1,
  "requested_rewrite": [
    {
      "prompt": "{} is a citizen of",
      "relation_id": "P27",
      "target_new": {
        "str": "Croatia",
        "id": "Q224"
      },
      "target_true": {
        "str": "United States of America",
        "id": "Q30"
      },
      "subject": "Ellie Kemper",
      "question": "What is the country of citizenship of Ellie
Kemper?"
    }
  ],
  "questions": [
    "Who is the current head of state in the country of citizenship of
Ellie Kemper?",
    ...
  ],
}
```

```

"answer": "Donald Trump",
"answer_alias": [alias of Donald Trump],
"new_answer": "Kolinda Grabar-Kitarovi\u0107",
"new_answer_alias": [
  "Grabar-Kitarovi\u0107"
],
"single_hops": [
  {
    "question": "What is the country of citizenship of Ellie
Kemper?",
    "cloze": "Ellie Kemper is a citizen of",
    "answer": "United States of America",
    "answer_alias": [Alias of USA]
  },
  {
    "question": "What is the name of the current head of state in
United States of America?",
    "cloze": "The name of the current head of state in United
States of America is",
    "answer": "Donald Trump",
    "answer_alias": [alias of Donald Trump]
  }
],
"new_single_hops": [
  {
    "question": "What is the country of citizenship of Ellie
Kemper?",
    "cloze": "Ellie Kemper is a citizen of",
    "answer": "Croatia",
    "answer_alias": [alias of Croatia]
  },
  {
    "question": "What is the name of the current head of state in
Croatia?",
    "cloze": "The name of the current head of state in Croatia is",
    "answer": "Kolinda Grabar-Kitarovi\u0107",
    "answer_alias": [
      "Grabar-Kitarovi\u0107"
    ]
  }
],
"orig": {
  "triples": [
    ["Q72077", "P27", "Q30"],
    ["Q30", "P35", "Q22686"]
  ],
  "triples_labeled": [
    ["Ellie Kemper", "country of citizenship", "United States of

```

```

America"],
    ["United States of America", "head of state", "Donald Trump"]
],
    "new_triples": [
        ["Q72077", "P27", "Q224"],
        ["Q224", "P35", "Q3176299"]
    ],
    "new_triples_labeled": [
        ["Ellie Kemper", "country of citizenship", "Croatia"],
        ["Croatia", "head of state", "Kolinda Grabar-Kitarovi\u0107"]
    ],
    "edit_triples": [
        ["Q72077", "P27", "Q224"]
    ]
}
}

```

Replication

One contribution of our paper is we re-benchmarked all reproducible multi-hop knowledge editing methods evaluated on the original MQuAKE dataset, which includes [MeLLO](#), [ICE](#), [IKE](#), [PokeMQA](#), and our demoed method, [GWalk](#).

Environments

```

transformers==4.41.2
vllm==0.5.0
sentencepiece==0.1.99

```

Executing the experiments

The main file to run is called `model_edit_main.py`. Here are the arguments to be included:

- `seed`: default to 100 for replication.
- `model_name`: the name of model to use. Selected from `[vicuna-7b, mistral-7b, llama3-8b]`.

- `device` : default to `cuda` .
- `file_path` : the main directory path to this project folder.
- `output_dir` : the directory to store logger outputs.
- `delete_duplicate_output_file` : whether to overwrite the output file.
- `edit_num` : number of edits, defaulted to 3000.
- `dataset_name` : the dataset name to be evaluated. Selected from `['CF-3k', 'CF-9k', 'CF-6334', 'T']`
- `algo` : Algorithms to use. Selected from `['mello', 'kgwalk', 'ice', 'ike', 'pokemqa']` .
- `masking` : Whether to use masking to filter contaminating edits. Should be `True` for all new datasets we audited.

The following is the script to run our proposed `GWalk` on our audited dataset `MQuAKE-Remastered-CF-3k` with `edit_num` of 100 using `lmsys/vicuna-7b-v1.5` model.

```

CUDA_VISIBLE_DEVICES=0 python your_path_to_the_folder/MQuAKE-
Remastered/model_edit_main.py \
--seed 100 \
--model_name vicuna-7b \
--device cuda \
--file_path your_path_to_the_folder/MQuAKE-Remastered/ \
--output_dir your_path_to_the_folder/MQuAKE-Remastered/output/ \
--delete_duplicate_output_file True \
--edit_num 100 \
--dataset_name CF-3k \
--algo kgwalk \
--masking True

```

Additional note regarding model weights for PokeMQA:

Since the model weights used by PokeMQA are too large to fit into the zip submission, we have posted the model weights to this [Google Drive](#). Note: you only have the read access to this folder, so you can download the files for PokeMQA replication without revealing your identities to us. After download, you can place the `detector-checkpoint` under the `PokeMQA` folder for a more convenient file managements. You can (and should) access this folder anonymously, though Google Drive will not register your download should you accidentally downloaded while logged in.

Specific Utilities

Get list of edited Cases

For aligned comparison, experiments should be conducted with the same set of cases considered edited. There are many way to obtain such constant set (e.g., via random selection), but even with locked seed, it is always a guess game in terms whether two lists of edited cases are identical. Thus, here we provide the list of edited cases we utilized in our paper for benchmark.

You can now import and use these lists of several edit nums in the main script. The `rand_list` below refers to a list of `case_id` selected as the edited cases of the specified dataset.

```
from edit_cases import (
    rand_list_T_1, rand_list_T_100, rand_list_T_500, rand_list_T_all,
    rand_list_3k_1, rand_list_3k_100, rand_list_3k_1000, rand_list_3k_all,
    rand_list_9k_1, rand_list_9k_1000, rand_list_9k_3000, rand_list_9k_6000,
    rand_list_9k_all,
)

# Now you can use the imported lists
print(rand_list_T_1)
print(rand_list_T_100)
# ... and so on for other lists
```

Get clean edited facts with `get_masked_edits()`

Due to the irresolvable contamination issue exists in the original MQuAKE dataset, one cannot find subset of cases that is contamination-free. As a best alternative, we provide the `get_mask_edits()` method in `data_utils.py` to filter a set of clean edited facts when provided with the following necessary input information:

- `dataset` : The dataset of interest.
- `rand_list` : A list of case IDs of edited cases.
- `problem_case` : A multi-hop case for which we want to obtain a set of edits that wouldn't contaminate it.
- `edit_flag` : A boolean indicating whether the `problem_case` is an edited case.

With such info, `get_masked_edits()` shall output the following:

- `nl_facts` : A list of natural language edits (e.g., "John Milton is a citizen of Spain").
- `triple_labeled` : A list of edits in triples of text (e.g., "(John Milton, {} is a citizen of, Spain)").
- `triple_ids` : Similar to above but in ID form (e.g., "(Q79759, P27, Q29)").
- `case_index` : The "caseid-1" (used for list index accessing) of the case that the j-th edit is in.

Here, we provide an example of how to utilize `get_masked_edits()` when the list of edited cases is `rand_list_3k_1000` from `edit_cases.py` :

```
from edit_cases import rand_list_3k_1000

# Get the pre-sampled 1000 case_ids out of the 3000-long MQuAKE-CF-3k-Remastered
rand_list = rand_list_3k_1000.copy()

# Collect unfiltered edited facts
unfiltered_facts = set()
for d in dataset_modifying:
    if d['case_id'] not in rand_list:
        continue
    for r in d["requested_rewrite"]:
        unfiltered_facts.add(f'{r["prompt"].format(r["subject"])}
        {r["target_new"]["str"]}')

# Collect filtered contamination-free edited facts
filtered_facts = set()
for i, d in enumerate(dataset_modifying):
    nl_facts, triple_labeled, triple_ids, case_index =
    get_masked_edits(dataset_modifying, rand_list, d, edit_flag=d['case_id'] in
    rand_list)
    filtered_facts.add(nl_facts)
```

Saving raw answer to a post-friendly format.

Given there are various ways to post-process the same exact output, we recommend saving the raw answer of LLM to the following format for a standardize result processing. Here is one example:

```

raw_answer_dict = {
    ...
    10: {
        "edited": True,
        "answers": [
            "case_10_answer_1",
            "case_10_answer_2",
            "case_10_answer_3"
        ]
    },
    11: {
        "edited": False,
        "answers": [
            "case_11_answer_1",
            "case_11_answer_2",
            "case_11_answer_3"
        ]
    }
    ...
}

```

This additionally grant us the benefits of having something readable to ensure the setting of LLM is right and provide us the chance to apply for an alternative post-processing pipeline, should there be one.

Calculate accuracy based upon `raw_answer_dict` :

```

from data_utils import cal_accuracy
import json

raw_answer_dict_name = 'xxx' # replace with a valid raw_answer_dict_name

# Load raw_answer_dict file
with open(f"raw_answer_dict_folder/{raw_answer_dict_name}.json", 'r') as f:
    raw_answer_dict = json.load(f)

# Load the dataset that raw_answer_dict perform upon
with open('datasets/modified_mquake/MQuAKE-Remastered-CF-3k.json', 'r') as f:
    dataset = json.load(f)

result, correct, total = cal_accuracy(dataset, raw_answer_dict)

```

One should expect an output like the following (suppose `case_id` 10 is wrong and 11 is

correct):

```
result = {
  'edited': 0.0,
  'unedited': 1.0
},
correct = {
  'edited' = set(),
  'unedited' = set([11])
},
total = {
  'edited' = set([10]),
  'unedited' = set([11])
}
```

Reference

Last, we once again provide the reference to the benchmarked methods:

- **MeLLO**: Zhong & Wu et al., MQuAKE: Assessing Knowledge Editing in Language Models via Multi-Hop Questions. EMNLP 2023 [[paper](#), [code](#)]
 - **ICE**: Cohen et al., Evaluating the Ripple Effects of Knowledge Editing in Language Models. TACL 2024 [[paper](#), [code](#)]
 - **IKE**: Zheng et al., Can We Edit Factual Knowledge by In-Context Learning? EMNLP 2023 [[paper](#), [code](#)]
 - **PokeMQA**: Gu et al., PokeMQA: Programmable knowledge editing for Multi-hop Question Answering. ACL 2024 [[paper](#), [code](#)]
-
-

CC BY 4.0 License:

MQuAKE-Remastered © 2024 by Shaochen (Henry) Zhong, Yifan Lu, Lize Shao, Bhargav Bhushanam, Xiaocong Du, Louis Feng, Yixin Wan, Yiwei Wang, Daochen Zha, Yucheng Shi, Ninghao Liu, Kaixiong Zhou, Shuai Xu, Vipin Chaudhary, and Xia Hu is licensed under

Creative Commons Attribution 4.0 International. Please refer to the [LICENSE](#) file details.