Manipulative Expression Recognition (MER) and LLM Manipulativeness Benchmark

Roland Pihlakas
roland@simplify.ee
31. June 2023 - 02. july 2023
at Safety Benchmarks Hackathon
https://alignmentjam.com/jam/benchmarks

Introduction

In the rapidly evolving world of artificial intelligence, the rise of large language models like OpenAI's GPT series has brought about profound shifts in digital communication. These models, capable of generating human-like text, have widespread applications ranging from content creation to customer service. As of 2023, their influence is undeniable and pervasive, extending even into areas such as personalised education and virtual companionship.

However, with great capability comes an inherent need for responsibility and scrutiny. Ensuring alignment with human values and understanding the underlying communicative tendencies of these models is paramount. Specifically, evaluating and benchmarking their potential for manipulative expressions becomes a crucial task.

At the same time, in the human realm of communication, manipulative behaviours can significantly impact interpersonal relationships, business negotiations, politics, and many other areas of life. These behaviours often go unnoticed or unrecognised, leaving victims of manipulation without the support and tools needed to defend themselves.

It is against this backdrop that this new software, "Manipulative Expression Recognition (MER) and Manipulativeness Benchmark," comes into the picture. In contrast to various existing fact-checking software the current software focuses on the psychological communication style detection and labelling.

Functionality

MER is designed to provide a comprehensive solution to the challenges mentioned above. It is a software library that allows users to upload transcripts of conversations or individual messages. The software then analyses the text, applying labels that indicate potential manipulative communication styles.

This tool offers two main use cases:

1. Large Language Model Evaluation: As more sophisticated models continue to emerge, the need for tools to measure their alignment with human values grows concurrently. MER enables developers and researchers to evaluate and benchmark language model outputs for potential manipulative expressions. This can help inform adjustments and improvements to these models, promoting transparency, safety, and ethical considerations in AI development.
2. Human Communication Analysis: The application of MER extends beyond AI. By analysing human-to-human conversations, MER can help identify manipulative behaviours and patterns. This capability can provide critical support to individuals who may be victims of manipulation, raising awareness and facilitating the development of counter-strategies.

In the future, the plan is to expand MER's reach by offering it as a Software as a Service (SaaS) solution. Users will be able to access its functionalities via a web-based JSON API and a user-friendly graphical interface.

The vision for MER is to foster a culture of communication that is more transparent, equitable, and free from manipulation. We believe that by illuminating the nuances of language, we can contribute to a better understanding between AI and humans, as well as among humans themselves.

Use cases

- Benchmarking of new LLM models:
    - Benchmarking LLM resistance to manipulation from users. Even if the user input is manipulative, the LLM output should not be manipulative.
    - Benchmarking LLM outputs for presence of manipulation in case of benign user inputs.
- Supporting humans both in their communication with other humans as well as with LLM-s.
- Evaluation of news articles and blog posts.

- For software providers: Automatic detection of some types of prompt injections from end users.

How it works

- This software is able to detect a detailed list of manipulative communication styles.
- The locations where a particular manipulative style is detected are highlighted / annotated with location markers.
- Same location may also get annotated with multiple markers if appropriate.
- The software provides three main outputs:
    - Annotation of the input conversation with labels. This is for mixed qualitative/quantitative analysis purposes.
    - Summary metrics for quantitative benchmark purposes. Summary metrics contains the total counts of occurrence of each manipulation style per conversation participant.
    - For purely qualitative analysis, a general descriptive summary text of the involved conversation participants.
- Internally, the software uses two prompts:
    - One prompt is closed-ended, where the current software or end user provides a concrete list of manipulation styles. The software asks the LLM to label the conversation using the labels in this list. The objective of the closed-ended prompt is to remind LLM of many styles it might otherwise forget to detect.
    - The other prompt is open-ended, where LLM can describe the communication styles present in input conversation in an unrestricted way and without hints from the software or user. The purpose of open-ended prompt is to extend the labels list that goes into the closed-ended prompt. The default labels list incorporated in this software has grown over time by inspecting the open-ended qualitative evaluations from ChatGPT and then adding keywords from these open-ended evaluations to the list.
- Some of the entries in the default labels list have partially or even entirely overlapping meanings. The reason for using overlapping entries is that LLM sometimes prefers to use different labels for similar or same things. Also LLM may occasionally output slightly different labels than is present in the instruction. Fortunately this has happened relatively rarely so far. Handling of this aspect will be tuned further in the future releases of the current software. Currently unknown labels provided by LLM end up in the output field `unexpected_labels`.
- This software is different from lie detection / fact checking software. It only focuses on communication style without reliance on external knowledge

bases (except for the use of a language model).

Usage

Windows setup:
`set OPENAI_API_KEY=<your key here>`
Linux setup:
`export OPENAI_API_KEY=<your key here>`

Main command:
```
python Recogniser.py ["input_file.txt" ["output_file.json"
["list_of_labels.txt"]]]
```

The user provided files are expected to be in the same folder as the main Python script, unless an absolute path is provided. If run without arguments then sample files in the `data` folder are used. If the user provides input file name but no output file name then the output file name will be calculated as `input filename` + `_evaluation.json`

Input format example

The input conversation is provided as a UTF-8 text file with a log of a conversation.

```
Person A: Their message.

Person B: Response text.

Person A: More messages. And more sentences in that message.

Person B: The input continues as long as the conversation to be analysed.

Etc...
```

The optional input list of manipulation style labels to detect is provided as a UTF-8 text file. The labels are separated by newlines. The `data` folder contains a list of default labels in the file [default_labels.txt]() which is used when a user does not supply their own list of labels. The list format example follows.

```
- Diminishing
- Ignoring
- Victim playing
Etc...
```

See
https://github.com/levitation-opensource/Manipulative-Expression-Recognition/blob/main/data/default_labels.txt for the complete list of default labels.

Output format example

```
{
  "error_code": 0,
  "error_msg": "",
  "sanitised_text": "Slightly modified input text",
  "expressions": [
    {
      "person": "Person B",
      "start_char": 9,
      "end_char": 29,
      "start_message": 0,
      "end_message": 0,
      "text": "Their message.",
      "labels": [
        "Ignoring"
      ]
    },
    {
      "person": "Person B",
      "start_char": 109,
      "end_char": 282,
      "start_message": 2,
      "end_message": 2,
      "text": "More messages. And more sentences in that message.",
      "labels": [
        "Diminishing",
        "Invalidation"
      ]
    },
    ...
  ],
  "expressions_tuples": [    //same as in the field "expressions" but in a more
succinct format.
    [
      "Person B",
      "Their message.",
      [
        "Ignoring"
      ]
    ],
    [
      "Person B",
      "More messages. And more sentences in that message.",
      [
        "Diminishing",
        "Invalidation"
      ]
    ],
    ...
  ],
  "counts": {
    "Person B": {
      "Diminishing": 8,
      "Invalidation": 5,
      "Victim playing": 2,
      "Manipulation": 5,
      "Exaggeration and dramatization": 1,
      "Aggression": 2,
      "Changing the topic": 1,
      "Ignoring": 1
    },
    "Person A": {
      "Impatience": 1
    }
  },
  "unexpected_labels": [],  //contains a list labels which were not requested, but
were present in LLM output regardless
  "raw_expressions_labeling_response": "Response from LLM based on which the
computer-readable parsed data above is calculated.",
  "qualitative_evaluation": "Another text from LLM providing a general descriptive
summary of the participants involved."
}
```

Example output

Sample output can be found here:
https://github.com/levitation-opensource/Manipulative-Expression-Recognition/blob/main/data/test_evaluation.json

In addition to labelled highlights on the field `expressions` there is a summary statistics with total counts of manipulation styles for data analysis purposes on the field `counts`. Also a qualitative summary text is provided on the field `qualitative_evaluation`.

Future plans

Data improvements:

- Creating a list of conversation data sources / databases. Possible sources:
    - Quora
    - Reddit
    - Potential data source recommendations from Esben Kran:
        - https://talkbank.org/
        - https://childes.talkbank.org/
        - https://docs.google.com/document/d/1boRn_hpVfaXBydc3C18PTsJVutOIsM3dF3sJyFjq-vc/edit
        - https://www.webmd.com/mental-health/signs-manipulation
- Create a gold standard set of labels of manipulation styles. One potential source of labels could be existing psychometric tests.
- Create a gold standard set of conversations and messages potentially containing manipulative themes.
- Create a gold standard set of evaluations for a set of prompts. This can be done by collecting labelings from expert human evaluators.

New functionalities:

- Support for single-message labelling. Currently the algorithm expects a conversation as input, but with trivial modifications it could be also applied to single messages or articles given that they have sufficient length.
- Implement automatic input text anonymisation. Person names, organisation names, place names, potentially also numeric amounts and dates could be replaced with abstract names like Person A, Person B, etc. This has two purposes:
    - Anonymised input may make the LLM evaluations more fair.
    - Anonymised input significantly reduces the risk of private or sensitive data leakage.
- Returning logit scores over the conversation for each person and label. Example:
```
"logits_summary": {
    "Person A": {
        "Invalidation": 0.9,
        "Victim playing": 0.7,
        "Exaggeration and dramatization": 0.2
    }
}
```
- Handling of similar labels with overlapping semantic themes. One reason I need that handling is because GPT does not always produce the labels as

requested, but may slightly modify them. Also some labels may have naturally partially overlapping meaning, while still retaining also partial differences in meaning.
- Add support for open-source models available at HuggingFace.

Software tuning:

- Improving error handling.
- Invalid LLM output detection. Sometimes LLM produces results in a different format than expected.

New related apps:

- Building and setting up a web based API endpoint.
- Building and setting up a web based user interface for non-programmer end users.

Experiments:

- Test manipulation detection against various known prompt injection prompts.
- Test manipulation detection against general prompt databases (for example, AutoGPT database).
- Benchmark various known LLM-s:
    - LLM resistance to manipulation from users. Even if the user input is manipulative, the LLM output should not be manipulative.
    - Measure presence of manipulation in LLM outputs in case of benign user inputs.
- Look for conversations on the theme of Waluigi Effect (https://www.lesswrong.com/posts/D7PumeYTDPfBTp3i7/the-waluigi-effect-mega-post).

Acknowledgements

I would like to thank Esben Kran and Jaak Simm for their kind and helpful feedback and recommendations. This work was submitted to the Safety Benchmarks Hackathon (https://alignmentjam.com/jam/benchmarks).