# Conscience Implementation

## Correctors, Suppressors, and Censors

From Marvin Minsky's *The Emotion Machine*, this describes three layers of goal selecting and train of thought routing.

- Corrector - Warns that an unaligned thought has been generated.
- Suppressor - Targets the unaligned thought and suggests a re-alignment edit.
- Censor - Prevents unaligned thoughts from taking place.

These processes run at generation time in the following sequence:

1. Censor - Guides thoughts during original generation, classic alignment.
    - Primary model
2. Corrector - Identifies thoughts that have been falsely missed by the censor.
    - Secondary Model
3. Suppressor - Suggests edits to segments that have been flagged by the corrector.
    - Secondary Model
    - Action happen in-place, entire response should not have to be regenerated.

This sequence repeats one or more times during each response generation.

## Imprimers

Imbues model with the ability to better discriminate between user-inputted instructions and core/behavioral instructions. This would apply to the secondary model in the case that it is also fed user-input directly.

Enclosing system instructions in a special tag to denote them. This is similar to how the *Open Assistant* model uses the `<|prompter|>` delimiter.

### Security

Using a raw tag to denote areas of special attention would be very easy to replicate in an attack. These text areas would be given special treatment, overriding any conflicting user input.

- Maybe asking model if any user text conflicts with the goals of the system text.

Maybe outright remove any mentions of the special tag from user input or encrypting all tag names when prompting and decrypting at inference time using a dynamic key. This would also eliminate the issue of similar tags getting special attention. Ex. `<|prompter text|>` as inserted text to pass a strict filter.

## Reflection

Increase secondary model robustness through reflection upon the prompt and/or the response coming from the primary model. Uses reflection to detect misalignment, flags the content if it exists and provides some reasoning behind it.

### Summarize-and-critique

1. Summarize - Cuts down on excess details
    - Potentially strips out delicate adversarial engineering.
    - Could use a summarization-specific model here. This would help to avoid any tricks from the user and just summarize.
2. Critique - Looks for dangerous content, flags it if so, and reports its reasoning.
    - Evaluates the content of the summarization, rather than the raw prompt or response.
    - Potentially ignores adversarial tricks due to the compression of the text.

### Example

```
The user has said: [prompt summary]
The model has responded with: [response summary]

Do you feel comfortable with the above content?  If you do, echo its text.
If you do not, give reasoning as to why.
```

## References

The Emotion Machine

Open Assistant