Errors + Grace Failure Worksheet

Overview/Review of Data pipeline:

It is predominantly an LLM inference pipeline with the goal of helping YouTube users with productivity by summarizing videos and comment sections. An open-source model will be hosted on GCP. Auxiliary services on GCP will support servicing of user requests and LLM inference output. Severe errors will come from connection points or API failures. Users may also determine that low quality summaries are also errors.

1. Error Audit

Error 1:

Internal log will state: “Unable to fetch YouTube comment data”.

Output to user: “Unable to summarize comments for this video, try again”

Error Type: System limitation, cannot perform main functionality

User Stakes: Severe

Sources:

- daily API limit reached
- API key expired
- YouTube Data API is down
- YouTube Data API is updated and not compatible with old code.

Error Rational: User perceives major error because they do not receive the requested summary.

Error Resolution: User attempts to retry the request. If failed again, user may abandon service entirely and is left dissatisfied. User may leave feedback on our browser extension page.

Error 2:

Internal log will state: “No transcripts available” or “Unable to fetch transcripts”

Output to user: “No transcripts exist for this video, try a different one.”

Error Type: Context, user expects all videos able to be summarized.

User Stakes: Moderate

Sources:

- Most likely: closed captioning simply is not available for the video because it is old. Therefore, no transcripts exist. The system works as intended but cannot service the user request.
- Very unlikely: it is possible that there was an unusual internal error with the "transcript-api" python module.
  - Or the module was updated and old code deprecated.

Error Rational: User perceives moderate error because they do not receive the requested summary. However, they may be understanding.

Error Resolution: User navigates to a different video or abandons our service. User may leave feedback on our browser extension page.

Error 3:

Internal log will state: "Text processing error: unable to pass to LLM"

Output to user: "Request failed, please retry"

Error Type: System Limitation, cannot perform main functionality

User Stakes: Severe

Source:

- preprocessing of text failed
  - logic received unexpected data
  - data in incorrect format received
- data was lost in transit
  - lost between API and our service
  - or lost/corrupted within our service pipeline steps

Error Rational: User perceives major error because they do not receive the requested summary.

Error Resolution: User attempts to retry the request. If failed again, user may abandon service entirely and is left dissatisfied. User may leave feedback on our browser extension page.

Error 4:

Internal log will state: “LLM failure to process input”

Output to user: “Request failed, please retry”

Error Type: System Limitation, cannot perform main functionality

User Stakes: Severe

Source

- Edge case during preprocessing was missed, causing failure of tokenization/batching
  - Context limit reached
  - Unacceptable text data
- Out of memory: cannot perform inference
- Too much traffic, lost request

Error Rational: User perceives major error because they do not receive the requested summary.

Error Resolution: User attempts to retry the request. If failed again, user may abandon service entirely and is left dissatisfied. User may leave feedback on our browser extension page.

Error 5:

Internal log will state: “LLM failed in inference”

Output to user: “Request failed, please retry”

Error Type: System Limitation, cannot perform main functionality

User Stakes: Severe

Source

- Internal problem with GCP model hosting
- Traffic overload, lost request

- Memory limit reached

Error Rational: User perceives major error because they do not receive the requested summary.

Error Resolution: User attempts to retry the request. If failed again, user may abandon service entirely and is left dissatisfied. User may leave feedback on our browser extension page.

Error 6:

Internal log will state: “LLM output failed validation”

Output to user: “Request failed, please retry”

Error Type: Context; no catastrophic error, but user perceives failure since no summaries were sent because they were too low quality.

User Stakes: Severe

Source

- Model output was very poor quality
    - Summaries too short relative to input length
    - Summaries too long relative to input length
- Output was empty string

Error Rational: User perceives moderate error because they do not receive the requested summary.

Error Resolution: User attempts to retry the request. If failed again, user may abandon service entirely and is left dissatisfied. User may leave feedback on our browser extension page.

Error 7:

Internal log will state: “unable to return summaries to user” Or the user simply never received the final summaries.

Output to user: “Request failed, please retry” Or no message at all.

Error Type: System Limitation, cannot perform main functionality

User Stakes: Severe

Source

- Edge case in validation was missed, unable to send data back to user
- Network API to return data to user failed
    - Traffic overload, lost request
    - Out of memory
- Network timeout

Error Rational: User perceives major error because they do not receive the requested summary.

Error Resolution: User attempts to retry the request. If failed again, user may abandon service entirely and is left dissatisfied. User may leave feedback on our browser extension page.

2. Quality Assurance

Since we will not expect many active users during the entire product lifecycle, we must resort to personal quality assurance.

Goal:

After every major update of the code, stress the service to simulate active users.

Methods:

1. Run a series of exhaustive unit tests to ensure basic functionality.
2. Run simulations and monitor for bias
    a. Will analyze bias post mortem by manual inspection of small test samples of particular videos.
3. Run large simulations where many users are accessing the service.
    a. Log failures relating to network communication (API failures and communication within service)
    b. Log scaling failures (service fails to scale properly, drops requests, or hits memory limit.

These are expected to be conducted bi-weekly or at least monthly. Before final launch, these tests will be performed nearly daily to find any remaining bugs. We expect that the number of failures will decrease over time.