Ping Pong Translate Cloud Computing Project on Google Cloud Platform+

Kacper Grzymkowski, Jakub Fołtyn, Rafał Klimek

Overview of app

This app allows uploading phrases from different languages. The phrase is then translated by a machine learning model into another language, and then the process is repeated multiple times (in a “ping pong” manner) translating the phrase from source language to target language and back. In the end the user can view how the phrase was distorted by multiple translations.

UI

The UI is a static web page hosted on Firebase Hosting. The logic is handled with plain JS. It does not have any server based UI components nor does it use any build engine. The landing page allows creating new “ping pongs”, by providing an original prompt, a source language and a target language. Upon successfully creating an element, a redirect is performed to an address with the appropriate ID, which is also used to keep track of the element ID. This page refreshes automatically every 0.5s, pulling new data from Firestore and updating the view. Once the ping pong changes status to PAUSED due to back-and-forth limit, the page also allows resuming of the game.

Storage

The prompts and completed “ping pongs” are saved to Firestore. This database requires a secondary index to speed up deduplication measures in the CreatePingPong Cloud function. Records (“ping pongs”) will be stored in the form of JSON files containing ID, date, original text passed by the user, languages of translations, status (PAUSED, RUNNING or LOOPED) and list of all the translations. Example records are shown on the diagram.

Processing

CreatePingPong Cloud Function is called by requests from the main page, when the user fills out the prompt. A new record is added to FireStore with status PAUSED.

The main processing module is dockerized and deployed using Cloud Run. It is invoked by CreatePingPong and ResumePingPong Cloud Functions, provided with the job’s ID.

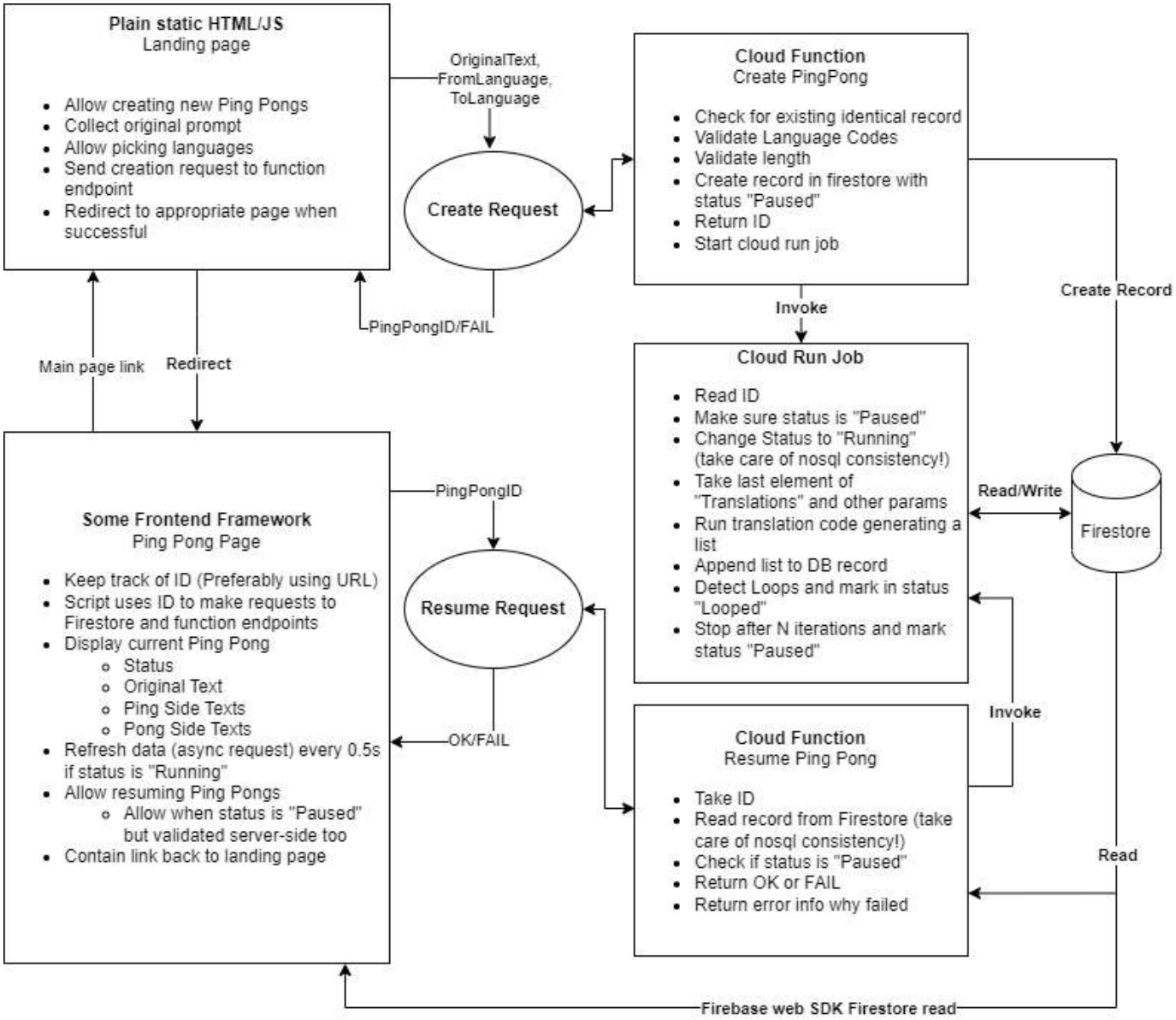
Its purpose is to run translating jobs on a requested record present in Firestore. It uses the NLLB model from HuggingFace. After the record is read, its status is changed to RUNNING

and iterations begin. After each successful translation the record is updated with new results. If the ping-pong enters a loop, status is changed to LOOPED and iteration is stopped. If 4 translations have been made (2 back-and-forths) the status is once again changed to PAUSED.

ResumePingPong Cloud Function is called by requests from the job details page on PAUSED jobs. The main processing module is called similarly to CreatePingPong.

Authorization

No user authorization. Firebase is open for outside read, write is reserved for the Cloud Function and Cloud Run.



Example Firestore Records

```
{
  "ID":"1234-abcd-...",
  "CreateDate": "2023-01-25T14:48:00.000Z",
  "OriginalText": "Hello World!",
  "FromLanguage": "Eng_Latn",
  "ToLanguage": "Pol_Latn",
  "Status": "Paused",
  "Translations": [
    "Hello World!",
    "Witaj, świecie!",
    "Hello, world!"
  ]
}
```

```
{
  "ID":"1234-abcd-...",
  "CreateDate": "2023-01-25T14:48:00.000Z",
  "OriginalText": "Hello World!",
  "FromLanguage": "Eng_Latn",
  "ToLanguage": "Pol_Latn",
  "Status": "Looped",
  "Translations": [
    "Hello World!",
    "Witaj Świecie!",
    "Hello World!"
  ]
}
```

```
{
  "ID":"1234-abcd-...",
  "CreateDate": "2023-01-25T14:48:00.000Z",
  "OriginalText": "Hello World!",
  "FromLanguage": "Eng_Latn",
  "ToLanguage": "Pol_Latn",
  "Status": "Paused",
  "Translations": [
    "Hello World!"
  ]
}
```