Homework 1

By Luca Comba

The repository can be found at https://github.com/lukfd/lab2_factories

To make it easier I created a Enum for all the classification strategies. I have then created a new Topic pydantic class. I have created a utility folder where I could store shared Exception classes and a utility library for reading and modifying any json file under the data directory.

I have then created a new `EmailSimilarityInferenceService` to inferr the similarity based on stored emails, and created a `InferenceServiceFactory` to enable a switch between the two services (similarity vs topic) based on the strategy selected.

New Endpoints

I implemented and updated the following API endpoints:

- `POST /emails`

This enpoint now stores a new email in the `data/emails.json`. The endpoint also accepts `subject`, `body`, and a optional `topic`. It then returns a message and the generated UUID email id.

- `POST /emails/classify`

This endpoint now classifies an email using a selectable strategy. The strategies can be `topic` (default), which compares the input email against topic descriptions, or `nearest` to finds the most similar labeled stored email and uses its topic.

- `POST /topic`

This endpoint is for adding a new topic. It takes a `name`, `description` as a json body request and saves it into the correct `topic_keywords.json` file.

New Class Inference

As previously mentioned, I wrote a `InferenceServiceFactory` to select the correct service (`topic` or `nearest`) based on request strategy.

The `EmailTopicInferenceService` handles topic-description based classification only, while the `EmailSimilarityInferenceService` handles nearest-email classification only.

This keeps each class focused and makes it easier to extend with more strategies later.

Testing

I added/updated tests for the new behavior under the `tests` directory. Run `make test` to run all the pytests.

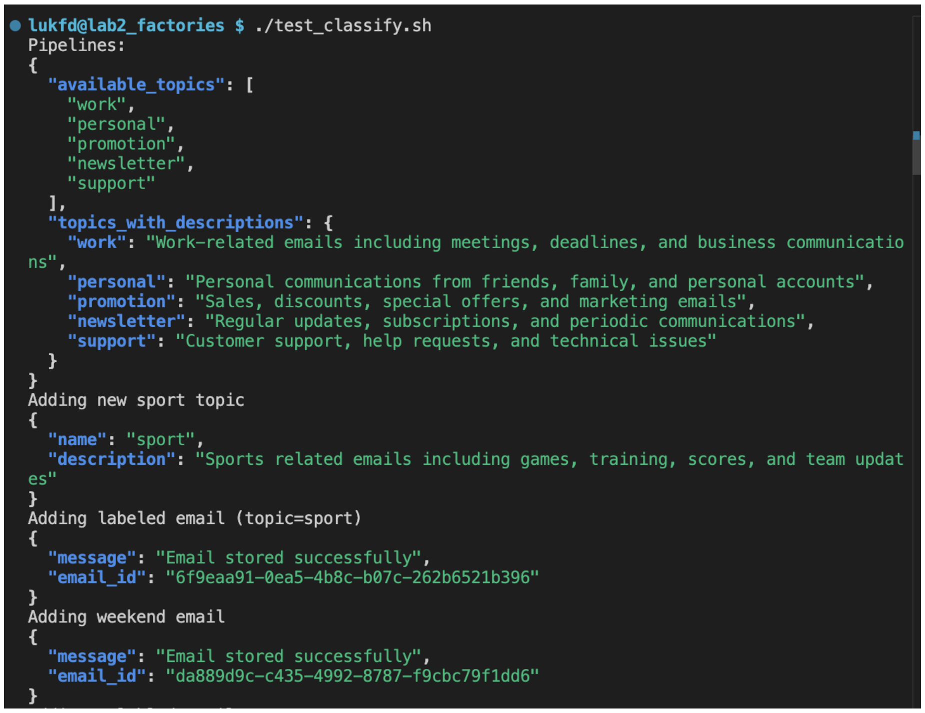
I have created a `test_classify.sh` to run a full manual API flow.

These output show that it was able to correctly add a new topic, add a new email.

To test the classification based on the `nearest` strategy, I have writted an email taking about the weekend and tagging it as `personal`. I then tried to classify a new email where I have mentioned the weekend. The model correctly predicted the email to be a personal email.

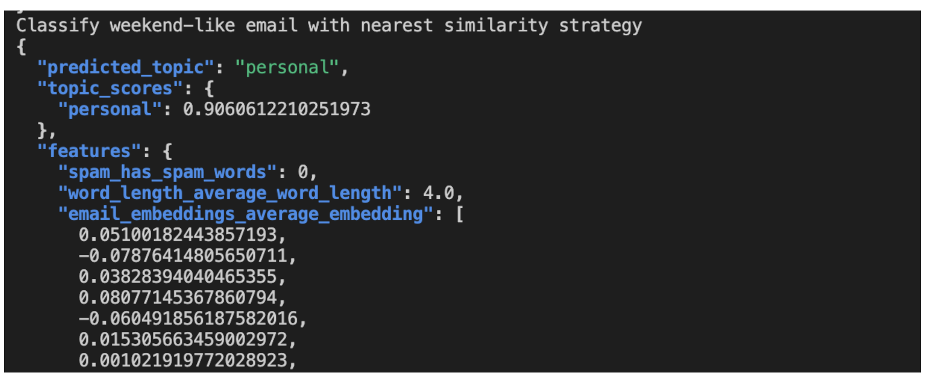
The `test_classify_output.txt` file contains all of the API responses and demonstrates the results of the classification.

Screenshots



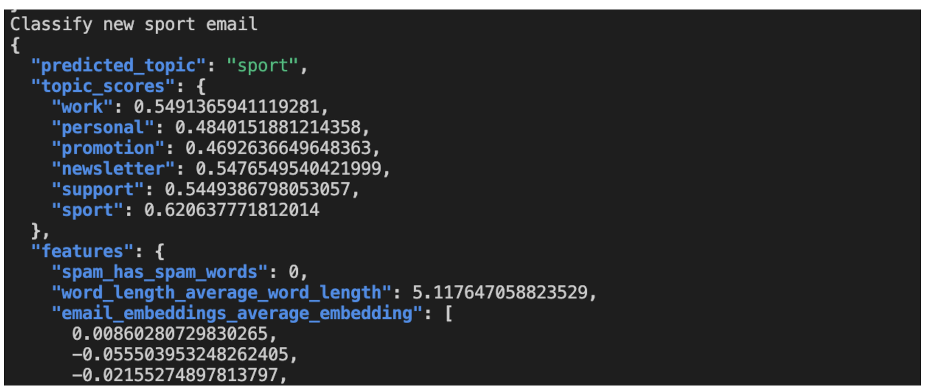
```
lukfd@lab2_factories $ ./test_classify.sh
Pipelines:
{
  "available_topics": [
    "work",
    "personal",
    "promotion",
    "newsletter",
    "support"
  ],
  "topics_with_descriptions": {
    "work": "Work-related emails including meetings, deadlines, and business communicatio
ns",
    "personal": "Personal communications from friends, family, and personal accounts",
    "promotion": "Sales, discounts, special offers, and marketing emails",
    "newsletter": "Regular updates, subscriptions, and periodic communications",
    "support": "Customer support, help requests, and technical issues"
  }
}
Adding new sport topic
{
  "name": "sport",
  "description": "Sports related emails including games, training, scores, and team updat
es"
}
Adding labeled email (topic=sport)
{
  "message": "Email stored successfully",
  "email_id": "6f9eaa91-0ea5-4b8c-b07c-262b6521b396"
}
Adding weekend email
{
  "message": "Email stored successfully",
  "email_id": "da889d9c-c435-4992-8787-f9cbc79f1dd6"
}
```

Figure 1: S1



```
Classify weekend-like email with nearest similarity strategy
{
  "predicted_topic": "personal",
  "topic_scores": {
    "personal": 0.9060612210251973
  },
  "features": {
    "spam_has_spam_words": 0,
    "word_length_average_word_length": 4.0,
    "email_embeddings_average_embedding": [
      0.05100182443857193,
      -0.07876414805650711,
      0.03828394040465355,
      0.08077145367860794,
      -0.060491856187582016,
      0.015305663459002972,
      0.001021919772028923,
```

Figure 2: S2



```
Classify new sport email
{
  "predicted_topic": "sport",
  "topic_scores": {
    "work": 0.5491365941119281,
    "personal": 0.4840151881214358,
    "promotion": 0.4692636649648363,
    "newsletter": 0.5476549540421999,
    "support": 0.5449386798053057,
    "sport": 0.620637771812014
  },
  "features": {
    "spam_has_spam_words": 0,
    "word_length_average_word_length": 5.117647058823529,
    "email_embeddings_average_embedding": [
      0.00860280729830265,
      -0.055503953248262405,
      -0.02155274897813797,
```

Figure 3: S3