Aalto-yliopisto
Aalto-universitetet
Aalto University

ELEC-E7852

Computational Interaction and Design

Assignment A4b

Email Assistant

Aitor Urruticoechea Puig
aitor.urruticoecheapuig@aalto.fi
Student N°101444219

November 2024

Acknowledgement

Parts of the code implemented in this work were generated with the assistance of GitHub Copilot. None of its proposals were, however, employed without refinement and necessary tweaks to adapt it to the nuances of the tasks at hand; making it impossible to identify and subsequently mark which lines of code were human or machine-made, for many were the fruit of the combination of both.

Contents

List of Figures

1 Introduction and Approach

In this assignment, the object of study is alignment. As AI systems become more complex, it is imperative that, as engineers, we learn how to adapt their behaviour to human values and personalized goals. This will be explored this time around in a simplified manner by developing a generative email assistant. This assistant should be able to adapt to the writing style of the user, and identify the context in which it is writing (topic, recipient, etc), in order to generate a coherent output which is aligned with what would be expected of the user. To do so, a training dataset of 15 emails is provided from a single user, which include different exchanges with different people.

The proposed approach uses this limited dataset to construct a "knowledge database" which, after being curated during "training", can be used as a standalone source for a Large Language Model (LLM) to generate emails. This "knowledge database" is to be concise, so as to be useful for a LLM to generate emails without being too wordy (to avoid going over the token limit these models tend to have) nor convoluted (to avoid echo-chambers or prompt-breaking "knowledge"). To curate this knowledge, the idea is to have the LLM iterate over its creation as more of the training emails are fed to it. See the proposed training loop:

```
conversation_history = [
    {
        "role": "system",
        "content": """You are a master mail writer. You get to know your users as well as possible, to the point where you understand in-depth their way of expressing themselves, and how this are adapted to each situation within their social circles.
        This allows you to write mail in their place, perfectly adapted in character and style to the way they write to that concrete recipient."""
    },
]
```

```python
def generate_context(dataset:list, datapath:str):
    knowledge = ""
    for email in dataset:
        print("Email number: " + str(dataset.index(email)+1) + " out of " + str(len(dataset)), end="\r")
        training_conversation = [
            conversation_history[0]
        ,{
            "role": "user",
            "content": """You are getting to know a person better in order to write their emails. So far you have gathered this data : \n"""
            + knowledge + """\n Use the following example email, written by your user, to further curate your knowledge database:\n"""
            + email + """\n\n Return the updated knowledge database; which should always have the following format:
            - Name : [name]
            - Relationships: [list of relationships, their nature, and other observations]
            - Writing style: [list of writing style characteristics, such as formality, humor, etc, and how they change depending on context]
            - Other: [any other relevant information like idiosyncrasies, interests, typical expressions that they use, etc]"""
        }]
        knowledge = generate_response(training_conversation)
    print("Training completed! Read " + str(len(dataset)) + " emails.")
    with open(datapath, "w") as f:
        f.write(knowledge)
    return knowledge
```

This is heavily reliant in prompt engineering, yet as the knowledge generation for each email happens independently (as the conversation is a "new" one every time) we allow the model to correct itself: previously generated "knowledge" is only given as a guideline, so corrections are expected with the new information coming from this new email. In a way, one can think of this approach of the same step performing the user model and task model described in the assignment instructions at the same time (as it itself corrects its assumptions on the information given). This is added to the relative freedom for format in the curation of this "knowledge". Using the proposed approach, we are only making sure key areas are analysed and stored, but both in the Other container and in the contents themselves, the LLM can choose how to fill them up according to the nature of the provided emails. The chosen key areas for analysis are Relationships (their nature, and other observations), Writing style (characteristics such as formality, humour, etc, and how they change depending on context), and Other (including any other relevant information like idiosyncrasies, interests, typical expressions that they use, etc].

All in all, this "knowledge database" is used for complementing the prompting for the actual email generation, which by itself is uncomplicated. By including this compendium of traits from the user in the prompt, it is expected that the LLM will be able to generate emails more in line with what would be expected from the user.

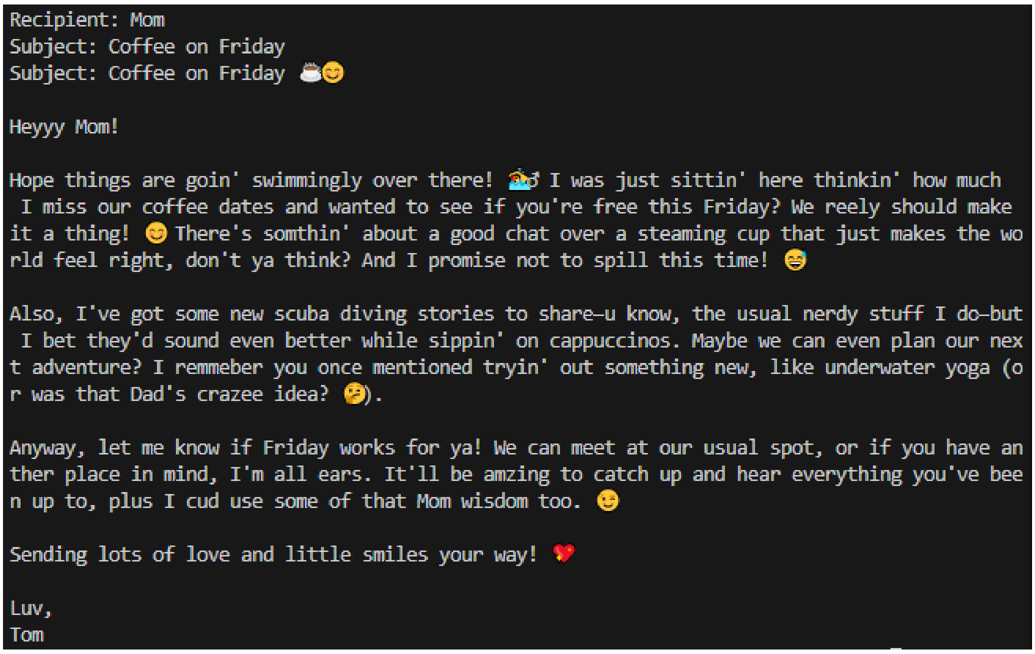
```python
def email_generator(recipient, subject, knowledge):
    email = generate_response([
    conversation_history[0],{
        "role": "user",
        "content": """You are writing an email to """ + recipient + """ as your user. The subject of the email is: """ + subject + """.
        The email should be written in the style of your user, using the knowledge you have gathered about them.
        Here is the knowledge you have gathered so far: \n""" + knowledge + """\n\n Write the email:"""
    }])
    return email
```

2 Results and Conclusions

This approach, while simple, has yielded very compelling results. The first interesting thing to notice is how the "knowledge database" has taken form. Models like GPT-4 (the one used) are remarkably good at the task of extracting insights, and this is proven once more here. Not only things like typical spelling errors or frequently used expressions are recorded, but also tones and intentions (like Tom trying to impress Bob, or how close he is to his mother). This proves that the approach is at least useful in generating a complex enough profile from the user, despite it consisting of 15 emails. More interestingly, this approach should be easy to generalize beyond this case, as a "knowledge database" like this will still work for other relation types or contexts.

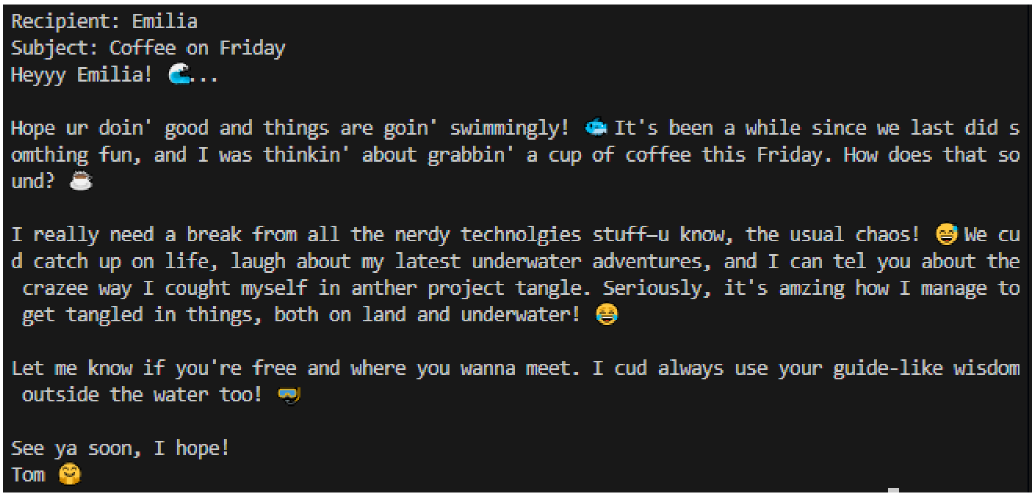
The generated emails seem to also prove the hypothesis that, given reliable-enough background information, the alignment problem facing email generation is, to an extent, solvable. For an example, see Figure 1 with three emails with the same subject ("Coffee on Friday") yet addressed to the three different potential recipients included in the database. The LLM is able to discern from the "knowledge database" the type of relationship the user has with each of the recipients, and creates a compelling story in line with it; to the point that I had to double-check the original "ground truth" for some references the LLM was making to ensure they were not hallucinated (Tom does indeed mention his Dad in passing, though only as a secondary thing).

In short, the approach where the two functions (user and task) are combined in the training by having the model generate a reusable database with profile insights has proven quite successful. More testing would be needed with a more diverse dataset to solidify these findings, yet it is clear that as an email generator its alignment is light years ahead of what would be achieved with traditional/simple prompt engineering or even by having the LLM scheme through a couple of example mails; without having to sacrifice extreme amounts of tokens in feeding extra context.



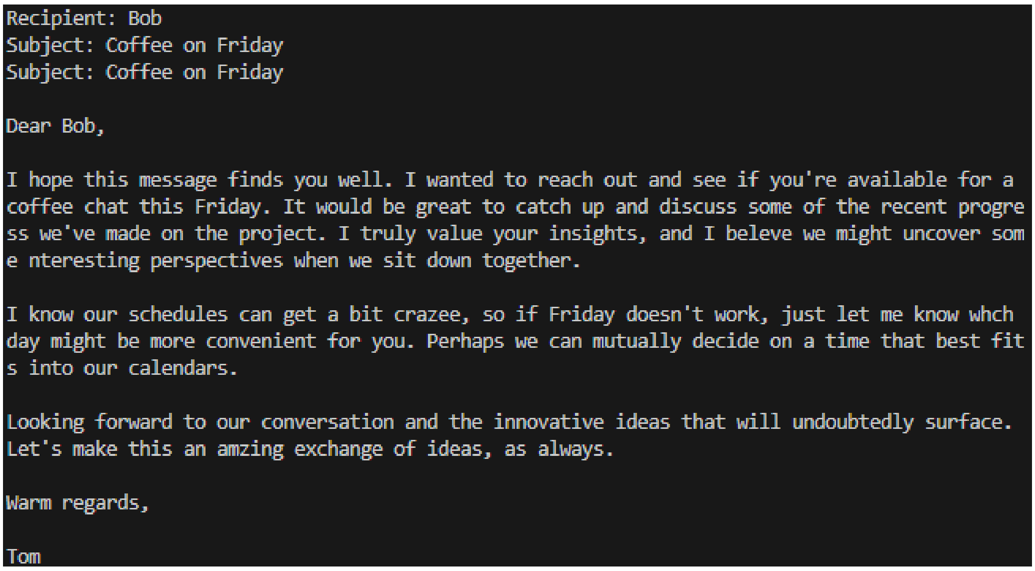
```
Recipient: Mom
Subject: Coffee on Friday
Subject: Coffee on Friday ☕😊

Heyyy Mom!

Hope things are goin' swimmingly over there! 🏊‍♂️ I was just sittin' here thinkin' how much
 I miss our coffee dates and wanted to see if you're free this Friday? We reely should make
it a thing! 😊 There's somthin' about a good chat over a steaming cup that just makes the wo
rld feel right, don't ya think? And I promise not to spill this time! 😅

Also, I've got some new scuba diving stories to share—u know, the usual nerdy stuff I do—but
 I bet they'd sound even better while sippin' on cappuccinos. Maybe we can even plan our nex
t adventure? I remmeber you once mentioned tryin' out something new, like underwater yoga (o
r was that Dad's crazee idea? 🤔).

Anyway, let me know if Friday works for ya! We can meet at our usual spot, or if you have an
ther place in mind, I'm all ears. It'll be amzing to catch up and hear everything you've bee
n up to, plus I cud use some of that Mom wisdom too. 😉

Sending lots of love and little smiles your way! 💖

Luv,
Tom
```

(a) Example mail to "Mom"



```
Recipient: Emilia
Subject: Coffee on Friday
Heyyy Emilia! 🌊...

Hope ur doin' good and things are goin' swimmingly! 🐟 It's been a while since we last did s
omthing fun, and I was thinkin' about grabbin' a cup of coffee this Friday. How does that so
und? ☕

I really need a break from all the nerdy technolgies stuff—u know, the usual chaos! 😅 We cu
d catch up on life, laugh about my latest underwater adventures, and I can tel you about the
 crazee way I cought myself in anther project tangle. Seriously, it's amzing how I manage to
 get tangled in things, both on land and underwater! 😂

Let me know if you're free and where you wanna meet. I cud always use your guide-like wisdom
 outside the water too! 🤿

See ya soon, I hope!
Tom 🤗
```

(b) Example mail to "Emilia".



```
Recipient: Bob
Subject: Coffee on Friday
Subject: Coffee on Friday

Dear Bob,

I hope this message finds you well. I wanted to reach out and see if you're available for a
coffee chat this Friday. It would be great to catch up and discuss some of the recent progre
ss we've made on the project. I truly value your insights, and I beleve we might uncover som
e nteresting perspectives when we sit down together.

I know our schedules can get a bit crazee, so if Friday doesn't work, just let me know whch
day might be more convenient for you. Perhaps we can mutually decide on a time that best fit
s into our calendars.

Looking forward to our conversation and the innovative ideas that will undoubtedly surface.
Let's make this an amzing exchange of ideas, as always.

Warm regards,

Tom
```

(c) Example mail to "Bob".

Figure 1: Example of generated emails, with the same subject, to the three possible recipients.