Project Report: Horoscope and Tarot App

Emelie Aalto & Nadina Suditu
Dialogue Systems 2 2022
University of Gothenburg

Introduction

Our original idea was making an app that would provide instructions and information on how to take care of various houseplants. However, we needed an API for that, and we couldn't find any on the Internet (the only one that had been available had been taken down). We thought about constructing our own API, but we decided it would have been impractical and very time consuming and ultimately irrelevant for our ultimate purpose: making a TDM app.

We arrived at creating a tarot and horoscope app after searching for APIs on topics that would match our interests, and thought it would be a bonus to use two of them together, since they're mutually relevant and would bring more depth and variety to the app. Technically, the main menu of the app introduces the user with 3 separate parts: horoscope, instructions on tarot readings, and information on all the cards in a classic tarot deck. Practically, the horoscope part is completely separate while the other two are intertwined and are meant to be used together.

The first part acts as a horoscope service: you can ask for your prognosis for the span of 3 days (yesterday, today, and tomorrow), for all of the 12 signs of the zodiac, for different categories. The second and third part are about tarot readings and what they mean.

Brief description of the pre-existing service and API (if applicable)

The aztro API https://aztro.sameerkumar.website/ is a free API which provides the horoscope with all the information in one piece, which we separated into chunks based on categories.

The Tarot API https://rws-cards-api.herokuapp.com/ is a Heroku based API which is basically a collection of the meanings, visual descriptions and all possible details about all of the cards in a tarot deck. From this, we only extracted the information that was relevant to our app: the practical meanings of the cards upright and reversed.

Data collection

For this, we split the task based on the app parts: Emelie collected the horoscope based dialogues and Nadina collected the tarot based dialogues. In total, 8 recorded dialogues were made, 4 for each part. Both parts had minimal instructions so as to not influence their dialogues too much and have them be as natural as possible. For the horoscope part one participant acted as the system and had access to the API where they could search for the data that the user participant asked. The user participant got provided beforehand with info on what the 12 zodiac signs are, and on what information they could ask from the system. This was the case for the tarot dialogues as well. The user knew that they could ask for information about 4 different types of spreads, and the system provided that information with the use of this site: https://www.alittlesparkofjoy.com/easy-tarot-spreads/

Horoscope dialogue example:

S: Yes. Hello, welcome to the horoscope app. What can I do for you today?
U: I want to know about my mom's zodiac sign.
S: What zodiac sign is your mom?
U: She's an aquarius.
S: Hmm. How is that spelled? Haha. Aquarius, okay I found it now. Here we go. Do you want to know the horoscope for today, or tomorrow?
U: Yesterday.
S: Yesterday. So the horoscope for aquarius for yesterday is "Oh sure! Just when you thought you were totally focused on what you were doing you were suddenly ready to ditch it all to have one perfect romantic moment. Lucky for you it's the right time."
U: What is the mood?
S: The mood. For yesterday?
U: Yes.
S: It was sweet.
U: Sweet... Which sign is she compatible with?
S: Aquarius was compatible with Leo yesterday.
U: Okay. Now I want to know about my sign.
S: What sign are you?
U: I'm a gemini.
S: You want your horoscope for yesterday, today or tomorrow?
U: Today.
S: Today. So your horoscope for today is "Fiery? Impulsive? Who? You. Well, not ordinarily but there are exceptions to every rule and you're due for one right now. This exception might last for a while so you might want to warn your friends now."
U: NO!
S: Yes it is.
U: What is my lucky number for today?
S: Your lucky number for today is 47. Anything else you want to know?
U: Ehm... color.
S: Your lucky color? For today? It is silver.
U: Okay, thank you. Good bye.
S: Thank you. Hope you have a nice evening.

Tarot dialogue example:

S: Hi, how can I help you?
U: I would like to know how to do the love cross
S: It is easy to do….it is…it's easy to do a love cross. First, the central card, or the theme, will stand for the present state of the issue between the quorant and the other person.
U: Yea, i got it.
S: Okay, good. Let's move to the second one.
U: Can I ask what the card means?
S: Sure
U: What does it mean? It's the eight of wands
S: The eight of wands represents rapid action, movement and quick decisions.
U: Thank you

S: Ok, now place the second one to the left of the theme, to represent the quarant's perspective.
U: What does the knight of wands mean?
S: The knight of wands action, adventure and fearlessness.
S: Now place the third card to the right of the theme, to show the other person's place.
U: Just…I know what it means. We can move on.
S: Great! The fourth card, placed below the central card, is the foundation of the relationship, or something in the past contributing to the current issue.
U: Mhm. Yes
S: Do you want to know…
U: No
S: Okay.
S: The next card…finally, the fifth card is placed above the central card, to show the likely outcome.
U: What does the 3 of wands mean?
S: The 3 of wands means looking ahead, expansion and rapid growth.
U: Thank you
S: You're welcome

When distilling the dialogues, we found that we didn't need to modify many parts, as usually the 'system' person was straight to the point, with a few hesitations here and there. As for our analysis of the dialogues, we discovered some behaviours that gave insight into what we should implement or modify in our app. For the horoscope, we found out that people tended to ask about information about one sign and day and then continue to ask more for categories for that same sign and day. Sometimes users didn't ask straight up what sign they wanted information on, e.g. in dialogue above "I want to know about my mom's zodiac sign" where the system has to ask again to get the name of the sign. For the tarot dialogues, the users tended to ask what a card meant immediately after putting it down, or let the system know that they already knew what it meant and that they could move on. Sometimes, the system would ask unprompted if the user wanted to know the meaning of the card. We will expand more on this later on in the report.

Implementation

The app is implemented using the TDM structure and what we learnt in the course. The app consists of two sections with multiple plans for the two sections. The horoscope section is only informational while the tarot section is both informational and instructional. For the horoscope part users can ask both incrementally and in one-shot for information about a particular horoscope. It can also handle over-answering and other-answering. Some examples:

- Incremental:

  U: "I want to know about my horoscope"
  S: "Which day?"
  U: "Today"
  S: "What sign?"
  U: "Leo"
  S: "What category?"
  U: "The mood"

S: “The mood is Happy”

- One-shot:

  U: “I want to know leos mood for today”

  S: “The mood is happy”

The user can also ask an optional explanatory question concerning what sign they are.
For the tarot parts, the menu allows the user to choose between going straight to the tarot card explanation, or to follow the spread instructions and ask about the cards at the same time. Therefore, we made it so that, at any point, the user can ask about the meaning of the card they are handling, to which the system asks what the name of the card is, and it redirects to the cards api, after which the dialogue goes back to the instructions. As mentioned above, for the informational parts we used two APIs, while for the instructional part we trained our model using Rasa. However, the app in its final form makes use of Rasa overall for interpreting the user answers, since we don’t interact with it in the TDM pipeline.

For the visual part, we exported the css code in a dedicated file and edited it according to our idea of what the app should look like. We added buttons for the main menu, as well as for the part in which the user can choose which tarot spread they want instructions on.

There’s a file in the “additional_files” folder in the repository named “test dialogues.pdf” with examples of dialogues handled by the system.

Discussion and possible future work

After collecting our dialogues for the horoscopes we noticed that people often switched topic whilst on the same main topic. This was something that we tried to implement in our app to make it more functional and more alike the way people talk about horoscopes. When the user and the system are talking about a horoscope for a sign for one day, they continue to ask for more categories on the same sign without having to tell the system participant that they still want information on the same sign and same day. This could be explained as the system forgetting the recent category but remembers the sign and the day and then search for the horoscope category for those. Since it occurred often in our recorded dialogues we thought this was important to implement and figured an easy solution would be for the system to forget the value for the category of horoscope but remember the other two.

However, using “forget_all” or “forget predicate” was not possible for this since the system could either forget all information given by the user or not remember any of it, it was not possible to only remember two out of three values. So we tried finding other ways of implementing this by making different plans, postplans, goals and assuming shared issues but that didn’t work either. The closest we got to implementing this function was by assigning the value manually, see code example below:

```
<assume_shared>
  <proposition predicate="sign_search" value="aries"/>
</assume_shared>
<assume_shared>
  <proposition predicate="day_search" value="tomorrow"/>
</assume_shared>
```

Here we assigned the sign to be “aries” and the day to be “tomorrow” and the system would give the horoscope for aries tomorrow. However it was not possible to assign the value to something other than an individual so this is not something that the app can handle at the moment. We then asked the lab assistants how to implement this in TDM and got the answer that it was not possible in TDM and that we should bring it up in the report. Unfortunately this made our app for the horoscope part very limited and not at all similar to the collected dialogues and is definitely a feature that could be added to TDM in the future.

Another thing we wanted to implement but we couldn’t in the way that we had imagined it, and had to find a workaround, was the feature of the user being able to move on from an instruction without any “report: done” utterance. This was concerning the part in which the system provides instructions on how to do a tarot spread, and the user wants to ask what a card means before the end of the plan. It was possible to do this, but after providing the card information, the system would go back to the previous instruction, making the dialogue unnatural and redundant. Unfortunately, it’s not possible to make the system move on without a “done” report as of now, so we just increased the time that the system has to wait for an utterance, and added more natural ways of saying “done” to the training data, to make the dialogue more seamless.

Topic switching is very limited in TDM, at least in the way we wanted it to work. Another case in which we were slumped by this was in the horoscope part, in which we would have liked to give the user the option of asking the system what their sign is by providing their birthday. This would have technically been possible with Python workarounds, but time consuming and in some ways irrelevant to the actual project goal, which was supposed to be working with what TDM can offer, not a Python assignment. In the lab we were provided with a ‘not ideal’ solution, in which the user can say that they don’t know their sign, and the system gives a prompt saying that they can guess a sign and then access the API for verifying the date range of that particular sign. We’re also aware that we could’ve made 366 individuals for each day of the year and then connect each date to their zodiac sign but it would also have been too time consuming, keeping in mind the fact that this was not a phenomenon that occurred in the dialogues, just something that we thought would be good if added. It is therefore a suggestion for the future.

We believe that our biggest struggle with TDM was related to topic switching, and also the fact that the documentation was very sparse, confusing, and oftentimes not provided in a good context. Everything that we mentioned above in this discussion would be features that could make our app better.