3D CHATBOT IN HIGHER EDUCATION, HELPING STUDENTS WITH PROCRASTINATION AND STUDY PLANNING PROBLEMS

K. Samyn
Howest University of Applied Sciences (BELGIUM)

Abstract

In our bachelor degree (Digital Arts and Entertainment) we have a lot of motivated and passionate students. However, the workload is high and it can become difficult for students to meet all the requirements of the study program and maintain a healthy work/life balance. The study program is open to all students and as a publicly funded institute we often lack the means to adequately track students and help them to reach their goals.

As a part of the solution, we are developing a chatbot solution that uses a 3D chatbot to help with the coaching of students. The virtual avatar integrates with the learning management system and uses elements of cognitive based therapy to help students overcome typical problems that plague our students such as procrastination, lack of study planning and communicative problems (with peers and staff). The chatbot should be intelligent enough to forward students to the correct persons if the system cannot determine the correct course of action.

The development of a 3D chatbot poses many problems and contains many moving parts. The technical implementation has support for 3D animation, face and emotion recognition, language understanding, speech recognition and text to speech.

The chatbot uses a simple pattern recognition system as a first step to recognize the intent of the user and extract entities. If no pattern is recognized, the user input is handed over to a third party language understanding service (e.g. Microsoft LUIS). The intent is then processed by our system which delivers the correct response. The subjects of the discussion are maintained in a DAG (directed acyclic graph) to make it possible to maintain a working state of the conversation. It is therefore possible for the student to reference an earlier response of the chatbot.

For the speech recognition and text to speech we also rely on third party services (i.e. Google Cloud speech). For the response model a programming language was developed that makes it possible to gives responses to the user (student) via multiple channels. The primary channel is of course text, which can be converted to speech, but it also possible to for example start a video in response to a request of the user. The coach bot can also receive cues about which animation to play as part of a response, or for example the facial expression that should be assumed.

Finally, the 3D chatbot can be hosted into any number of platforms including mobile which makes it possible for students to access the chatbot whenever needed.

Keywords: coaching, chatbot, 3D.

1 INTRODUCTION

The development of a chatbot is a multidisciplinary effort. In the first part of the paper we will focus on the technical analysis and requirements of the chatbot, and in the second part we will focus on the development process of the chatbot.

The technical analysis indicates that a major part of the chatbot solution is the capability to maintain the state of the conversation and to resolve referential questions to enable an actual conversation. A couple of existing chatbot technologies such as AIML[1] and Rivescript[2] were evaluated, however these were found to be lacking in certain areas that are critical for the development of our own chatbot system. Another critical area is the integration with a language understanding[3] tool such as LUIS or DialogFlow, given that AIML and Rivescript use simple pattern matching to understand the meaning of the user. Finally, the chatbot has a 3D virtual avatar representation for which the Unity game was chosen and which leverages the speech recognition services that were developed by Google (i.e. Google Cloud Speech).
The development of the chatbot itself then requires a multidisciplinary approach where educators, pedagogues and psychologists can contribute their expertise. A major problem for this research project is the translation of psychological models into meaningful conversation that have an evidence based benefit to the student. In other words, it is a tough and labour intensive problem to translate human expertise into a chatbot.

2 METHODOLOGY

The development of the chat was divided into two phases. The goal of the first phase was to develop a prototype that can be used to test various approaches to the development of the 3D chatbot. The goal of the second phase is to test the chatbot and add content and possibilities to the chatbot. In this second phase we also look at various approaches and models to develop the chatbot further.

2.1 Development of the chatbot prototype (phase 1)

Before the start of the project, the stakeholders determined the necessary components that would have to be present in the chatbot. From this requirement analysis it was apparent that there were a large number of unknown parameters that would need to be tested before the system could be rolled out. The prototype therefore had to support various features in various ways. The following table gives an overview of the different parameters that can be configured to test aspects of the chatbot system:

<table>
<thead>
<tr>
<th>Variant A</th>
<th>Variant B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Speech recognition (google cloud speech)</td>
</tr>
<tr>
<td>Webcam (emotion detection)</td>
<td>No emotion detection</td>
</tr>
<tr>
<td>Avatar</td>
<td>3D avatar</td>
</tr>
<tr>
<td>Chatbot interactivity</td>
<td>Autonomous</td>
</tr>
<tr>
<td>Output</td>
<td>Text to speech (google cloud speech)</td>
</tr>
</tbody>
</table>

A general overview of the components of the chatbot are given in the following figure:

![Figure 1. Chatbot technical overview.](image)

2.2 Content of the chatbot (phase 2)

For the second phase we analyzed a couple of models that can be implemented in the chatbot. A first model is the coaching model for e-learning by Salomon [5], which offers a useful model that divides the capabilities of a coaching system in several tiers. The first tier of the system is a practical tier that helps students with questions such as:

- Who is my coach?
- How do I file an assignment?
- Where is my next class?

These practical questions model reasonably well to a simple chatbot system, although an integration with the LMS (learning management system) is necessary. The final tier of this coaching model provides...
the support and response that are necessary to help students with study planning problems. In a way, the chatbot system has to build a ground layer of trust by providing helpful responses to typical practical problems, before students can entrust the system with more personal problems.

Another model that will be used in addition to the Salomon model is the open learner model [6]. In this model the student is an active agent and not a passive receiver of stimuli. The student actively processes the external stimuli and adjusts his internal strategy on this basis of his own interpretation of these stimuli. This interpretation is not necessarily the correct interpretation but improves the autonomy of the student and in turn the results of this student. The open learner model defines four phases, where every phase is defined according to the COPES principle (Cognitive conditions, Operations, Product and Evaluations). The chatbot focuses on the cognitive conditions (what is the interpretation of the phases) and the product (what is the end result of the student) to improve the evaluation.

3 RESULTS

The 3D chatbot was developed in the Unity game engine. Unity offers a simple development environment for 3D and offers sufficient capabilities to interface with the various cloud services that are used in the prototype.

3.1 Prototype

3.1.1 Input from user

The user can interface with the system via a chat window where the user can type and via speech. For the speech recognition and text to speech service (covered further) we opted for the Google Cloud Speech service. This choice was mainly governed by the requirement that the chatbot should understand the Dutch and English language and also be able to output Dutch and English via text to speech. The Dutch voice of the Google Cloud speech service is implemented via Wavenet[4] which offers a low cost realistic Dutch voice. A technical hurdle is the speech detection or VAD (voice activation detection). It is necessary within the game engine that accurately detect when the user is speaking. For this purpose a custom voice activation detection algorithm was developed that analyzes the speech patterns of the user and labels the microphone input so that a sentence can be send to the Google cloud speech service.

Another input type is the webcam where the user can be detected and the emotion of the user can be analyzed. The emotion of the user can then be used as an extra input to the chatbot system.

3.1.2 Language understanding

The language understanding component translates the text or voice input of the user to a recognizable intent that can be used to determine the response of the chatbot system. Language understanding works by providing a couple of example sentence of what the user might say to enable the tool to “understand” many variations of the same concept. For example, the following sentences all match the intent problem.course and extract an entity course from the sentence:

Table 2. Example intents and entity.

<table>
<thead>
<tr>
<th>Example sentence</th>
<th>Matched intent</th>
<th>Matched entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>I have problems with the drawing course.</td>
<td>problem.course</td>
<td>drawing</td>
</tr>
<tr>
<td>I don’t understand math</td>
<td>problem.course</td>
<td>Math</td>
</tr>
<tr>
<td>I am failing at 3D modelling</td>
<td>problem.course</td>
<td>3D modelling</td>
</tr>
</tbody>
</table>

An additional component in our proposed solution is the addition of simple patterns which can help to streamline the development process. An advantage of the system is that the costs associated with using a webservice such as Microsoft LUIS can be mitigated. When a sentence is entered into the system, an attempt will be made to extract the intent and entities via simple pattern matching. If this attempt fails the sentence will be transferred to the language understanding service, which will return an intent or entity. If the language understanding tool fails, the chat responds with a default message that it does not understand the user.
3.1.3 Chatbot logic

After an intent has been detected, the chatbot needs to respond with an appropriate message. From our literature study it was clear that existing chatbot solutions are not data-driven, in other words, the responses from the chatbot are coded mainly as literal text nodes that are defined within the chatbot system. I propose another arrangement where the knowledge of the system is stored in a knowledge graph that is sampled in response to input from the user. In figure 3, a simple example of such a knowledge graph is shown:

![Knowledge Graph Diagram](image)

The nodes in such a knowledge graph are called entities and these entities can form relationships with other entities. When the user asks for a specific bit of information, the chatbot system can query the knowledge graph for this information and create a response based on the data in the knowledge graph. This ensures that the data for the chatbot system can be altered independently of the chatbot system.

To make this possible, a programming language named Cobble was developed to efficiently create response to a specific intent and to make it possible to query data from a knowledge graph. An example of the cobble language will be presented in the following paragraph.

An advantage of using a knowledge graph is that the chatbot can maintain a list of topic (i.e. nodes) that the user has talked about. A typical scenario that is enabled by this system is the following:

**User**: Who is my coach?

**Chatbot**: Your coach is Alexandra Mores?
User: Where can I find her?

Chatbot: She is currently teaching in room L.101. Shall I make an appointment for you?

A referential word such as “her” is very natural for a user to use and it is a common expectation in conversations that such phrases will be understood by the chatbot. By maintaining a list of subjects and a data driven approach, such a contextual conversation becomes possible.

### 3.1.4 Natural language generation

Most chatbots use literal texts in response to a request from the user. A data drive approach offers a possibility to create template responses that adapt to the contents of the knowledge graph. As shown in Figure 4, the user asks for a specific person and the information about this person is queried via the knowledge graph. Literal text can then be collated together with information from the knowledge graph to provide a response.

```java
@graph.teach.single $teach {
  r 1 = @graph.person.reference
    + " teaches the course "
    + $teach + ". "
;
}
```

![Figure 4. Natural language generation.](image)

Finally, an extra feature of the system is the concept of channels where extra information about the response can be defined. For example, the response to a question could include a link to a video (online or within the LMS). A response could also define that the chatbot needs to show a certain animation (e.g. waving for goodbye, nodding in agreement, looking down when embarrassed, ...). A typical example is the difference between the literal text that can be show in the chat window and the text that must be sent to the TTS system, which includes hints such as pauses, breaks, emphasis and support for abbreviations.

### 3.1.5 Output

Finally, the text output is presented to the user and the audio output of the text to speech system is played to the user. The text is first converted to an SML (speech markup language) representation before it is converted to audio with the help of Google Cloud speech.

### 3.2 Further chatbot development

The next phase in the development will try to gather as much feedback from users as possible. The first hurdle to take is to gain the trust of the users (i.e. students) that the system can be useful.

### 4 CONCLUSIONS

The development of a 3D avatar chatbot system is a huge undertaking. It is a labour intensive work to come to the point where different methodologies and models can be tested with the end user. We hope
to present a basic system that can be further tested and developed and will prove to be a platform for the development of advanced coaching chatbot systems.

It is definitely the case that the development of a competent chatbot system requires a multidisciplinary team where equal importance must be given to the technical development and the content development of the system.

REFERENCES