BRIAN WELLS
OMSCS 6460 SUMMER 2017
HISTABOT 0.1 PROPOSAL
June 17, 2017

OVERVIEW

1. Project Background and Description

During the Spring 2017 semester, Brodrick Stigall completed a development project as a part of OMSCS 6460 that resulted in a conversational intelligent tutoring system (CITS) for the Google Home titled Histabot. In Histabot, Mr. Stigall created a conversational agent capable of imitating George Washington with the intent of teaching history to elementary school aged children. In its current form (which I have labelled Histabot 0.0), the agent can respond to common questions from students with a historically accurate (for the most part) answer with a first-person narrative as if George Washington was speaking himself. Histabot is also capable of making suggestions to students of other questions to ask if the agent is posed a question it does not have the answer to. To provide a more interactive experience, specific common questions that elementary aged children typically ask, yet where no historical information is available, were provided anecdotal responses (e.g. asking 'what is your favorite color?' results in the response 'That's an easy one. My favorite colors are America's colors: Red, White, and Blue!') (Stigall, 2017).

The scope of Mr. Stigall's project allowed for the hardcoding of specific responses to questions for George Washington. I am proposing to continue Mr. Stigall's efforts on Histabot to incorporate the ability for the agent to consume automatically generated responses for a larger set of historical figures. Instead of hard-coding responses to a particular set of questions, the goal of this project is to allow Histabot to automatically consume data from the internet to generate responses for at least 20 additional historical figures for students to converse with. This extension of Histabot will be called Histabot 0.1.

2. Motivation

The motivation for this proposal to extend Histabot 0.0 is multifaceted. Firstly, the field of CITS appears to be relatively unexplored, and the concept of a CITS being used to teach history from a first person perspective seems to be novel. Most of the CITS in use today focus on tutoring subjects in the science, technology, engineering, and mathematics (STEM) fields. The reasoning behind this makes sense, as these fields often involve difficult concepts that require knowledge construction for understanding, as opposed to learning the subject of history, which mostly involves committing facts to memory.

Another motivation behind this proposal is the fact that Histabot leverages relatively new technology in the form of Google Home. Early research suggests that spoken language tutoring provides an advantage over written (Pon-Barry, Clark, Owen Bratt, Schultz, & Peters, 2004), however the technology to support human-like dialogue is just now becoming prevalent. Products like Google Home and Amazon Echo are at the leading edge of speech recognition and speech-to-text technologies, providing developers with application programming interfaces to build sophisticated conversational applications. Leveraging these technologies to build a CITS could be the future of tutoring.

3. Project Scope

The scope of Histabot 0.1 includes the evaluation of YodaQA, an open-source question answering artificially intelligent agent. YodaQA is modelled after IBM's question answering agent Watson, along with cutting edge

research in the natural-language processing and information extraction fields (YodaQA, n.d.). The evaluation of YodaQA will involve creating a development instance of the agent and testing the accuracy of responses in the field of history.

If the accuracy of YodaQA is deemed sufficient, a production instance of YodaQA as a web-based application programming interface (API) will be created. A middleware layer will then be added to Histabot to interface with the YodaQA API. The middleware will convert the personalized questions directed to Histabot into a format suitable for YodaQA, and then convert the responses back from YodaQA into a first-person response to the user. For example, if the student asks Histabot, "when were you born?", and the current context is that the student is interviewing George Washington, the middleware will convert the question to "what was George Washington born?" Subsequently, if the response from YodaQA is "George Washington was born on February 22, 1732," the middleware will convert the response to "I was born on January 1st, 1999."

As a contingency if the accuracy of YodaQA is unacceptable, an independent data collection application will be developed. The data collection application will be capable of consuming data from different data sources and converting that data into a database that will be integrated into Histabot. The schema of the database shall conform the specifications of the current technology being used by Histabot. An example of a potential data source for the data collection application is Wikipedia.

Provided the development of the independent data collection application is necessary, a spiraled approach will be taken. The first spiral will involve the extraction of answers from the data source to predefined questions, such as "when were you born?" The data collection application will find all the answers possible to the provided questions and convert the answers to a response told from a first-person perspective (e.g. "I was born in 1984").

If time allows, a second spiral will include the extraction of subject specific questions and answers. For example, the application will find the profession of the subject historical figure and search for relevant questions pertaining to that profession, such as providing the answer to "who was your vice president?" if the subject is a president.

4. Deliverables

Schedule

Deliverable	Delivery Date
Five weekly status reports detailing progress, challenges, and revised expectations	Every week on Sunday from 25 June 2017 to 23 July 2017
Intermediate milestone 1	02 July 2017
Intermediate milestone 2	16 July 2017
Final paper	30 July 2017
Video presentation	30 July 2017
Source code	30 July 2017

Intermediate Milestone 1

For the first intermediate milestone, one of two possible deliverables will be generated depending on the accuracy of YodaQA. If YodaQA proves to sufficiently accurate, then a web-based API to YodaQA will be exposed to the public and a URL will be provided. The API will consist of a single endpoint that accepts a question as a request and the response is YodaQA's answer to the question. Alternatively, if the accuracy of YodaQA proves to be insufficient, a design of the information extraction and personalization pipeline will be outlined. The design will include technologies to be used as well as detailed diagrams depicting the flow of information.

Intermediate Milestone 2

As with the first milestone, the second milestone will depend on the accuracy of YodaQA. If the accuracy of YodaQA proves sufficient, then the web-based API from the first milestone will be exposed again, this time with the inclusion of a middleware layer to convert the questions and answers into a conversational format and will require the requests to

contain the context of a historical figure. Alternatively, if the accuracy of YodaQA proves to be insufficient, a video presentation will be created that demonstrates the automatic database generation pipeline from the GATE information extraction, to the first-person conversation formatted question/response schema.

5. Implementation Plan

Summary

As outlined in the Project Scope, the initial thrust in the implementation of Histabot 0.1 will involve the evaluation of YodaQA. This evaluation will consist of the creation of a development instance of YodaQA that leverages Wikipedia as its knowledge base. Once the development instance of YodaQA is operational, an evaluation of the accuracy of YodaQA will ensue. The evaluation will consist of manually asking YodaQA a series of questions posed in a variety of grammatical formats. For example, "when was George Washington born?" and "in what year was George Washington born?" are two similar questions yet are formatted quite differently. In the interest of time, an incremental approach will be taken when posing questions to YodaQA. Initially, 20 questions will be posed to two well-known historical figures. If YodaQA provides correct responses to greater than 18 (90%[1]) of these questions, the evaluation will continue. Next, 100 questions will be posed to YodaQA, ten questions for ten different historical figures. If the agent maintains a 90% accuracy rate, another round of 100 questions will be tested. If YodaQA passes all three of these rounds with 90% or greater accuracy, the agent will be deemed acceptable and the Histabot will be modified to incorporate it. Otherwise, the implementation of Histabot 0.1 will continue with the development of the independent data collection tool.

Provided that YodaQA fails the accuracy evaluation, an information extraction technology will be leveraged to build a question and answer database for a number of historical figures. GATE (General Architecture for Text Engineering), a well-documented, open source tool for text processing, is the proposed technology that will be used for information extraction (GATE Information Extraction, n.d.). Wikipedia articles will provide the document source for GATE. In order to extract information from Wikipedia, a set of question types will need to be defined and fed into the GATE pipeline. Once the information has been extracted using GATE for the historical figures, it will then be translated into the existing question/answer database schema currently being used in Histabot 0.0. This schema is a JSON (JavaScript Object Notation) format that can be recognized by api.ai, the conversational user experience platform currently used by Histabot (api.ai, n.d.).

Task List

YodaQA Integration

Task	Target Completion Date
Spin up instance of YodaQA targeting Wikipedia	22 June 2017
Generate first round of YodaQA evaluation questions	23 June 2017
Perform initial YodaQA evaluation	24 June 2017
Generate second round of YodaQA evaluation questions	27 June 2017
Perform second YodaQA evaluation	30 June 2017
Intermediate Milestone 1	02 July 2017
Generate final round of YodaQA evaluation questions	07 July 2017
Perform final YodaQA evaluation	10 July 2017
Intermediate Milestone 2	16 July 2017
Integrate YodaQA into Histabot pipeline	25 July 2017

---

3 [1] *It should be noted that a cursory research into Human-Computer Interaction was performed in an attempt to find any guideline or rule-of-thumb to follow for an acceptable level of accuracy. None was found.*

Task	Target Completion Date
Write final paper	29 July 2017
Create video presentation	30 July 2017
Package source code for delivery	30 July 2017

GATE Database Construction (assuming second round evaluation failure)

Task	Target Completion Date
Spin up instance of YodaQA targeting Wikipedia	22 June 2017
Generate first round of YodaQA evaluation questions	23 June 2017
Perform initial YodaQA evaluation	24 June 2017
Generate second round of YodaQA evaluation questions	27 June 2017
Perform second YodaQA evaluation	30 June 2017
Intermediate Milestone 1	02 July 2017
Generate subset of list of target information to construct database	04 July 2017
Generate GATE resources for target information extraction	08 July 2017
Extract information for subset of target information	09 July 2017
Automate database generation from extracted information	14 July 2017
Intermediate Milestone 2	16 July 2017
Generate remaining list of target information to construct database	22 July 2017
Extract remaining information	23 July 2017
Integrate new database into Histabot	27 July 2017
Write final paper	29 July 2017
Create video presentation	30 July 2017
Package source code for delivery	30 July 2017

6. Works Cited

(n.d.). Retrieved from api.ai: https://api.ai/

GATE Information Extraction. (n.d.). Retrieved from GATE: https://gate.ac.uk/ie/

Pon-Barry, H., Clark, B., Owen Bratt, E., Schultz, K., & Peters, S. (2004). Evaluating the Effectiveness of SCoT: A Spoken Conversational Tutor.

Stigall, B. (2017). *Histabot. A unique historical tutor.*

YodaQA. (n.d.). Retrieved from GitHub: https://github.com/brmson/yodaqa